

PEDRO CARVALHO ROCHA

**A FORMA DO SOM:
UM OLHAR APROFUNDADO SOBRE A REPRESENTAÇÃO GRÁFICA DE
ONDAS SONORAS**

Dissertação apresentada à
Universidade Federal de Viçosa, como
parte das exigências do Programa de
Pós-Graduação em Biologia Animal,
para obtenção do título de *Magister
Scientiae*.

VIÇOSA
MINAS GERAIS – BRASIL
2018

**Ficha catalográfica preparada pela Biblioteca Central da Universidade
Federal de Viçosa - Câmpus Viçosa**

T

R672f
2018

Rocha, Pedro Carvalho, 1990-
A forma do som : um olhar aprofundado sobre a
representação gráfica de ondas sonoras / Pedro Carvalho Rocha.
– Viçosa, MG, 2018.
xx, 188f. : il. (algumas color.) ; 29 cm.

Inclui apêndices.

Orientador: Pedro Seyferth Ribeiro Romano.

Dissertação (mestrado) - Universidade Federal de Viçosa.

Inclui bibliografia.

1. Bioacústica. 2. Bioinformática. 3. Anfíbios.

I. Universidade Federal de Viçosa. Departamento de Biologia
Animal. Programa de Pós-Graduação em Biologia Animal.

II. Título.

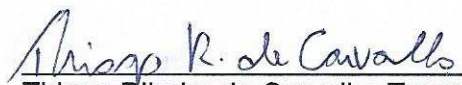
CDD 22. ed. 591.594

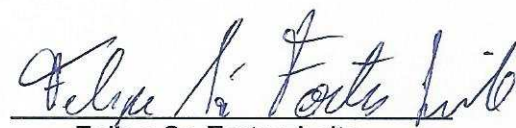
PEDRO CARVALHO ROCHA

**A FORMA DO SOM:
UM OLHAR APROFUNDADO SOBRE A REPRESENTAÇÃO GRÁFICA
DE ONDAS SONORAS**

Dissertação apresentada à Universidade Federal de Viçosa, como parte das exigências do Programa de Pós-Graduação em Biologia Animal, para obtenção do título de *Magister Scientiae*.

APROVADA: 13 de abril de 2018.


Thiago Ribeiro de Carvalho Tavares


Felipe Sa Fortes Leite


Pedro Seyferth Ribeiro Romano
(Orientador)

*Dedico este trabalho ao meu falecido avô Cyro,
que acreditou em mim quando eu ainda usava fraldas.*

*E também a meus pais, avós e amigos,
pois ainda continuam acreditando.*

À Carla, pois, sem ela, nem a metade seria possível.

AGRADECIMENTOS

Primeiramente aos meus pais, **Fernando** e **Liana**, meu irmão **Rafael**, minhas avós **Glória** e **Mirinha**, e a **Pretinha**, por todo o suporte, amor e compreensão.

À **Carla**, minha amada, pela melhor parte de cada dia, todos os dias. (Sem falar no companheirismo!)

Ao professor e orientador **Pedro Romano**, que aceitou me guiar nessa empreitada.

Ao estimado professor e amigo **Renato Feio**, que tanto me ajudou e orientou ao longo do mestrado, mesmo sem fazer parte desse projeto.

Um *Muito obrigado* a toda a comunidade de programadores da plataforma **R** pela inestimável ajuda nas horas de desespero.

Aos **Fritos**, por toda a amizade e loucura recreativa.

À turma da capoeira **Cordão de Ouro**, pela companhia e pela mandinga.

Aos amigos **Douglas** e **Jonas** pela inestimável ajuda nas montanhas.

A toda a turma do **Museu de Zoologia João Moojen** pela imensa receptividade e pelo ambiente agradabilíssimo de trabalho. Show!!

Também à turma do Laboratório de Osteologia e Paleontologia Comparada (**LAPOC**) pelo companheirismo e pelas tantas seções de café e prosa.

Agradeço ao Ministério do Meio Ambiente (MMA) e Instituto Chico Mendes de Conservação da Biodiversidade (ICM-Bio) pela licença de coleta nº 56090-1.

Ao Instituto Estadual de Florestas pela licença de coletas nº 100/2016.

À Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) pela bolsa de mestrado.

SUMÁRIO

LISTA DE FIGURAS.....	vi
LISTA DE TABELAS.....	xvii
LISTA DE ABREVIACÕES.....	xviii
RESUMO.....	xix
ABSTRACT.....	xx
ARTIGO CIENTÍFICO – Rocha, P.C. & Romano, P.S.R. The Shape of Sound: Two new $\mathbb{R}$ functions that translate sound waves into point configurations. <i>Methods in Ecology and Evolution.</i>1	
1. ABSTRACT	2
2. KEYWORDS.....	3
3. INTRODUCTION	3
4. MATERIALS AND METHODS.....	5
4.1 NEW METHOD: <i>SOUND SHAPE ANALYSIS</i>	7
4.2 ADDENDUM TO <i>EIGENSOUND ANALYSIS</i>	9
4.3 GEOMETRIC MORPHOMETRIC ANALYSIS.....	10
5. EMPIRICAL STUDIES: RESULTS AND DISCUSSION.....	13
5.1 SOUND SHAPE VARIATION BETWEEN EIGHT <i>PHYSALAEMUS</i> SPECIES	13
5.2 SOUND SHAPE VARIATION BETWEEN THREE POPULATIONS OF <i>PHYSALAEMUS DEIMATICUS</i>	18
6. CONCLUSIONS	22
7. ACKNOWLEDGEMENTS	24
8. CONFLICT OF INTEREST.....	24
9. AUTHOR'S CONTRIBUTIONS.....	24

10. DATA ACCESSIBILITY.....	24
11. REFERENCES.....	25
12. APPENDIX 1.....	30
13. APPENDIX 2.....	34
14. APPENDIX 3.....	180
15. APPENDIX 4.....	186

LISTA DE FIGURAS

ARTIGO

Figura 1. Spectrogram outputs generated by *SoundShape* function to represent stereotyped advertisement calls (i.e. sound units) of 30 *Physalaemus* species. Spectrograms were colored above the curve of relative amplitude at -25 dB, with red dots representing two-dimensional semilandmarks for each call. A) Sound window (left) and scale of amplitude colors (right) employed on spectrogram outputs. B) Color legend of major clades as proposed by Lourenço et al. (2015). Species marked with asterisks were employed on intraspecific empirical study and *Physalaemys deimaticus* (**) was employed on interpopulational study. Tree topology adapted from Lourenço et al. (2015).....6

Figure 2. Schematic steps in *SoundShape* (A–D) and *eigensound* (A, E–G). A) Both analysis are conducted from sound waves. *SoundShape*: B) A 2D spectrogram output is created for each wave file on a given folder. C) The longest CRA is computed at -25 dB. D) Point coordinates are acquired by cross correlation between the CRA and frequency values representing energy quantiles. *Eigensound*: E) Function computes 3D spectrograms for each wave file. F) A sampling grid is superimposed onto the sound surface, resulting in a simplified spectrogram. G) The sampling grid is used to acquire 3D coordinates that represent the shape of sound. 2D and 3D sets of coordinates are subject to Generalized Procrustes Analysis (GPA) and then Principal Components Analysis (PCA) subsequently.9

Figure 3. Semilandmark (SM) configurations acquired from the sample of 30 *Physalaemus* species and subject to *SoundShape* (A-D) and *eigensound* (E-F) approaches. A) Complete set of unaligned SM coordinates; B) SM coordinates after Generalized Procrustes Analysis (GPA); C) SM coordinates after GPA using a minimum Procrustes distance D) and a minimum bending energy criterion for sliding SMs; E) Complete set of unaligned SM coordinates; F) SM coordinates after GPA. In A-D, hollow circles represent the complete set (individuals) of 16 SMs and orange circles represent mean coordinates. In E-F, amplitude scale in colors is given.12

Figure 4. Ordination analysis of *SoundShape* data. A) Set of SM acquired by *SoundShape* function from the calls of eight *Physalaemus* species. Hollow circles represent SM after GPA with minimum Procrustes distance criterion for sliding SMs. Orange dots represent consensus shape. B) Eigenvalues as percentage of variance (grey bars) and broken stick model (red line and dots) computed from PCA using

Procrustes residuals. C) Individual projection of *Physalaemus* species onto PC1 and PC2. Each point represent one note sampled from individuals and convex hulls comprise all notes analyzed for each species. Semilandmarks (dots) and deformation grids in relation to consensus configuration represent minimum and maximum deformations from PCs and are displayed at the respective edges of ordination plot. Legend represent colors used for each species. D) Unvouchered males of *Physalaemus* species. Top left: *P. cuvieri*; top right: *P. erythros*; center: *P. rupestris*; bottom left: *P. signifer*; bottom right: *P. nattereri*. Photos from F. Leal and L.O. Drummond.15

Figure 5. Ordination analysis of *eigenSound* data. A) Set of semilandmarks (SM) acquired by *eigenSound* function from the calls of eight *Physalaemus* species. SM points are colored in relation to amplitude values. B) Eigenvalues as percentage of variance (grey bars) and broken stick model (red line and dots) computed from PCA using Procrustes residuals. C) Individual projection of *Physalaemus* species onto PC1 and PC2. Each point represent one note sampled from individuals and convex hulls comprise all notes analyzed for each species. Hypothetical SM configurations representing minimum and maximum deformations from PCs are displayed at the respective edges of ordination plot. Legend represent colors used for each species. D) Unvouchered males of *Physalaemus* species. Top left: *P. cuvieri*; top right: *P. erythros*; center: *P. rupestris*; bottom left: *P. signifer*; bottom right: *P. nattereri*. Photos from F. Leal and L.O. Drummond.....17

Figure 6. Ordination analysis of *SoundShape* data. A) Set of SM acquired by *SoundShape* function using the calls of *Physalaemus deimaticus* from three populations. Hollow circles represent SM after GPA with minimum Procrustes distance criterion for sliding SMs. Orange dots represent consensus shape. B) Eigenvalues as percentage of variance (grey bars) and broken stick model (red line and dots) computed from PCA using Procrustes residuals. C) Individual projection of *P. deimaticus* specimens onto PC1 and PC2. Each point represent one sampled note and convex hulls comprise all notes analyzed for each population. Semilandmarks (dots) and deformation grids in relation to consensus configuration represent minimum and maximum deformations from PCs and are displayed at the respective edges of ordination plot. Legend represent colors used for each individual and localities encompassed by convex hulls. Ind = individual. Loc = locality. D) Unvouchered male of *P. deimaticus* from Serra do Cipó, Morro do Pilar Municipality. Photo from L.O. Drummond.....20

Figure 7. Ordination analysis of *eigenSound* data. A) Set of semilandmarks (SM) acquired by *eigenSound* function using the calls of *Physalaemus deimaticus* from three populations. SM points are colored in relation to amplitude values. B) Eigenvalues as percentage of variance (grey bars) and broken stick model (red line and dots) computed

from PCA using Procrustes coordinates. C) Individual projection of *P. deimaticus* specimens onto PC1 and PC2. Each point represent one sampled note and convex hulls comprise all notes analyzed for each population. Hypothetical SM configurations representing minimum and maximum deformations from PCs are displayed at the respective edges of ordination plot. Legend represent colors used for each individual and localities encompassed by convex hulls. Ind = individual. Loc = locality. D) Unvouchered male of *P. deimaticus* from Serra do Cipó, Morro do Pilar Municipality. Photo from L.O. Drummond22

APÊNDICE (APPENDIX) 2

Figure 1. Spectrogram (above) and oscillogram (below) views of the advertisement call of *Physalaemus centralis*.40

Figure 2. Spectrogram (above) and oscillogram (below) views of the advertisement call of *Physalaemus kroyeri*.40

Figure 3. Standard two dimensional spectrogram of the advertisement call of *Physalaemus centralis* (left) and the three dimensional equivalent of it (right).....41

Figure 4. Standard two dimensional spectrogram of the advertisement call of *Physalaemus kroyeri* (left) and the three dimensional equivalent of it (right).....41

Figure 5. Two (left) and three dimensional spectrogram views (right) of the advertisement call of *Physalaemus centralis* with relative amplitude contours at -25 dB.43

Figure 6. Two (left) and three dimensional spectrogram views (right) of the advertisement call of *Physalaemus kroyeri* with relative amplitude contours at -25 dB.44

Figure 7. Spectrogram view of the advertisement call of *Physalaemus centralis* with longest curve of relative amplitude at -25 dB and square limits drawn around contour.47

Figure 8. Spectrogram view of the advertisement call of *Physalaemus kroyeri* with longest curve of relative amplitude at -25 dB and square limits drawn around contour.48

Figure 9. Spectrogram view of the advertisement call of <i>Physalaemus centralis</i> with longest curve of relative amplitude at -25 dB and two semilandmarks (red dots) plotted using minimum (SM-min) and maximum (SM-max) frequency values within the curve.	50
Figure 10. Spectrogram view of the advertisement call of <i>Physalaemus kroyeri</i> with longest curve of relative amplitude at -25 dB and two semilandmarks (red dots) plotted using minimum (SM-min) and maximum (SM-max) frequency values within the curve.	51
Figure 11. Spectrogram view of the advertisement call of <i>Physalaemus centralis</i> with longest curve of relative amplitude at -25 dB, minimum (SM-min) and maximum (SM-max) frequency values, and semilandmarks plotted as cross correlation between contour and frequency value representing 50% energy quantile. Semilandmarks represented by red dots.	57
Figure 12. Spectrogram view of the advertisement call of <i>Physalaemus kroyeri</i> with longest curve of relative amplitude at -25 dB, minimum (SM-min) and maximum (SM-max) frequency values, and semilandmarks plotted as cross correlation between contour and frequency value representing 50% energy quantile. Semilandmarks represented by red dots.	58
Figure 13. Spectrogram view of the advertisement call of <i>Physalaemus centralis</i> with longest curve of relative amplitude at -25 dB, SM-min and SM-max, and 14 semilandmarks resulting from the cross correlation between contour and seven frequency values representing energy quantiles. Semilandmarks represented by red dots.	61
Figure 14. Spectrogram view of the advertisement call of <i>Physalaemus kroyeri</i> with longest curve of relative amplitude at -25 dB, SM-min and SM-max, and 14 semilandmarks resulting from the cross correlation between contour and seven frequency values representing energy quantiles. Semilandmarks represented by red dots.	62
Figure 15. Spectrogram view of the advertisement call of <i>Physalaemus centralis</i> with complete set of 16 semilandmarks plotted as cross correlation between longest curve of relative amplitude at -25 dB, and frequency values representing energy quantiles within the contour. Semilandmarks represented by red dots.	65
Figure 16. Spectrogram view of the advertisement call of <i>Physalaemus kroyeri</i> with complete set of 16 semilandmarks plotted as cross correlation between longest curve of	

relative amplitude at -25 dB, and frequency values representing energy quantiles within the contour. Semilandmarks represented by red dots.....66

Figure 17. Complete set of 16 semilandmarks (red dots) and wireframe representing the sound shape of the advertisement call of *Physalaemus centralis*. Semilandmarks were acquired through cross correlation between longest curve of relative amplitude and frequency values representing energy quantiles within the contour.67

Figure 18. Complete set of 16 semilandmarks (red dots) and wireframe representing the sound shape of the advertisement call of *Physalaemus kroyeri*. Semilandmarks were acquired through cross correlation between longest curve of relative amplitude and frequency values representing energy quantiles within the contour.68

Figure 19. (Displayed below) Spectrogram views of the advertisement calls of 30 *Physalaemus* species with complete set of 16 semilandmarks acquired as cross correlation between longest curve of relative amplitude at -25 dB, and frequency values representing energy quantiles within the contour. Each species had a representative call/note selected previously and spectrograms were generated automatically by SoundShape function.79

Figure 20. Spectrogram view of the advertisement calls of *Physalaemus centralis* (left) and *P. kroyeri* (right) computed with default medium-sized window length $wl=512$88

Figure 21. Spectrogram view of the advertisement calls of *Physalaemus centralis* (left) and *P. kroyeri* (right) computed with small-sized window length $wl=256$89

Figure 22. Spectrogram view of the advertisement calls of *Physalaemus centralis* (left) and *P. kroyeri* (right) computed with default large-sized window length $wl=1024$90

Figure 23. Spectrogram view of the advertisement calls of *Physalaemus centralis* (left) and *P. kroyeri* (right) computed with little overlap $ovlp=40$ between successive windows.91

Figure 24. Spectrogram view of the advertisement calls of *Physalaemus centralis* (left) and *P. kroyeri* (right) computed with little overlap $ovlp=60$ between successive windows.92

Figure 25. Spectrogram view of the advertisement calls of <i>Physalaemus centralis</i> (left) and <i>P. kroyeri</i> (right) computed with little overlap $ovlp=80$ between successive windows.	93
Figure 26. Spectrogram view of the advertisement call of <i>Physalaemus centralis</i> placed at time = 0 s.	94
Figure 27. Spectrogram view of the advertisement call of <i>Physalaemus kroyeri</i> placed at time = 0 s.	95
Figure 28. Simplified grid of semilandmark coordinates with 100 cells on the time X-axis and 80 cells on the frequency Y-axis, enabling the acquisition of 8000 semilandmarks from the sound surface.	97
Figure 29. Three dimensional spectrograms resulting from the simplified sampling grid applied to the advertisement calls of <i>Physalaemus centralis</i> (left) and <i>P. kroyeri</i> (right).	98
Figure 30. Simplified grid of semilandmark coordinates using a logarithmic sequence on the time X-axis. Grid encompass 100 cells on the time X-axis and 80 cells on the frequency Y-axis, enabling the acquisition of 8000 semilandmarks from the sound surface.	99
Figure 31. Three dimensional spectrograms resulting from the simplified sampling grid using a logarithmic sequence on the time X-axis, and applied to the advertisement calls of <i>Physalaemus centralis</i> (left) and <i>P. kroyeri</i> (right).	100
Figure 32. Three dimensional view of spectrograms as points, representing the semilandmarks acquired from a linear sampling grid. Advertisement calls of <i>Physalaemus centralis</i> (left) and <i>P. kroyeri</i> (right).	102
Figure 33. Three dimensional view of spectrograms as points. Each point represents a semilandmark acquired from a logarithmic sampling grid. Advertisement calls of <i>Physalaemus centralis</i> (left) and <i>P. kroyeri</i> (right).	103
Figure 34. (Displayed below) Three dimensional view of spectrograms as points. Each point represents a semilandmark acquired from a logarithmic sampling grid and each plot is an stereotyped advertisement call of 30 <i>Physalaemus</i> species. These species had a representative call/note selected previously and spectrograms were generated automatically by the <code>eigensound</code> function.	112
Figure 35. Complete set of semilandmarks (hollow circles) acquired by <code>SoundShape</code> function from representative advertisement calls of 30 <i>Physalaemus</i> species. Orange	

dots represent mean coordinates for each of the 16 semilandmarks calculated from the complete set of semilandmark coordinates.118

Figure 36. Left: Complete set of three dimensional semilandmarks acquired by `eigensound` function from representative advertisement calls of 30 *Physalaemus* species. Right: Mean coordinates calculated from the set of three dimensional semilandmarks.119

Figure 37. Complete set of semilandmarks (hollow circles) after Generalized Procrustes Analysis. Orange dots represent consensus shape calculated from aligned semilandmarks. Semilandmarks acquired by `SoundShape` function using representative advertisement calls of 30 *Physalaemus* species.121

Figure 38. Left: Complete set of three dimensional semilandmarks after Generalized Procrustes Analysis. Right: Consensus shape calculated from the aligned set of three dimensional semilandmarks. Semilandmarks acquired by `eigensound` function using representative advertisement calls of 30 *Physalaemus* species.....122

Figure 39. Complete set of semilandmarks (hollow circles) after Generalized Procrustes Analysis using minimum Procrustes distance (left) and minimum bending energy criteria (right). Orange dots represent consensus shape calculated from aligned semilandmarks. Semilandmarks acquired by `SoundShape` function using representative advertisement calls of 30 *Physalaemus* species.124

Figure 40. Eigenvalues as percentage of variance (grey bars) and broken stick model (red dots and line) computed from Principal Components analysis using Procrustes coordinates. Data acquired using `SoundShape` function and 2D residuals after GPA + sliding.127

Figure 41. Eigenvalues as percentage of variance (grey bars) and broken stick model (red dots and line) computed from Principal Components analysis using (right) original coordinates acquired by `eigensound` function (left) and Procrustes coordinates after GPA (right).128

Figure 42. Individual projection of 30 *Physalaemus* species onto PC1 and PC2 axes using Procrustes coordinates acquired by `SoundShape` function. Numbers are colored according to phylogenetic groups from Lourenço et al. (2015) and convex hulls encompass major clades within *Physalaemus* (for numbers relative to species, groups and clades, see previous code chunk and Rocha & Romano, *in prep*, Fig. 1). Legend represent colors for each *Physalaemus* groups and clades.133

Figure 43. Individual projection of 30 *Physalaemus* species onto PC1 and PC2 axes using Procrustes coordinates acquired by `eigenSound` function. Numbers are colored according to phylogenetic groups from Lourenço et al. (2015) and convex hulls encompass major clades within *Physalaemus* (for numbers relative to species, groups and clades, see previous code chunk and Rocha & Romano, *in prep*, Fig. 1). Legend represent colors for each *Physalaemus* groups and clades.....134

Figure 44. Semilandmarks (dots) and deformation grids of minimum (left) and maximum (right) scores of the first Principal Component (PC1; 62.9% of variance). Deformation grids are given in relation to consensus configuration (i.e. score 0 of PC1) calculated from Procrustes coordinates acquired by `SoundShape` function. Data gathered from a representative advertisement call of 30 *Physalaemus* species.....137

Figure 45. Semilandmarks (dots) and deformation grids of minimum (left) and maximum (right) scores of the second Principal Component (PC2; 14% of variance). Deformation grids are given in relation to consensus configuration (i.e. score 0 of PC2) calculated from Procrustes coordinates acquired by `SoundShape` function. Data gathered from a representative advertisement call of 30 *Physalaemus* species.138

Figure 46. Hypothetical semilandmark configurations of minimum (left) and maximum (right) scores of the first Principal Component (PC1; 10.6% of variance) calculated from Procrustes coordinates acquired by `eigenSound` function. Data gathered from a representative advertisement call of 30 *Physalaemus* species.....139

Figure 47. Hypothetical semilandmark configurations of minimum (left) and maximum (right) scores of the second Principal Component (PC1; 10.4% of variance) calculated from Procrustes coordinates acquired by `eigenSound` function. Data gathered from a representative advertisement call of 30 *Physalaemus* species.140

Figure 48. (Displayed below) Spectrogram views of the advertisement calls of eight *Physalaemus* species including the set of 16 semilandmarks acquired from the curve of relative amplitude at -25 dB. Each species had a representative note selected previously. Spectrograms were generated automatically by `SoundShape` function.145

Figure 49. Aligned semilandmarks acquired by `SoundShape` function using advertisement calls of eight *Physalaemus* species. Hollow circles represent complete set of SMs after Generalized Procrustes Analysis using minimum Procrustes distance. Orange dots represent consensus shape calculated from Procrustes coordinates....148

Figure 50. Eigenvalues as percentage of variance (grey bars) and broken stick model (red line and dots) computed from PCA using Procrustes coordinates. Data acquired by

SoundShape function using the advertisement call of eight *Physalaemus* species and 2D residuals after GPA + sliding.....149

Figure 51. Individual projection of eight *Physalaemus* species onto PC1 and PC2 using Procrustes coordinates acquired by SoundShape function. Each point represent one sampled note and convex hulls comprise all notes analyzed for each species. Legend represent colors used for each species.151

Figure 52. Semilandmarks (dots) and deformation grids in relation to consensus configuration (i.e. PC score = 0). Grids calculated using Procrustes coordinates acquired by SoundShape function. Data from the advertisement call of eight *Physalaemus* species. Upper panel: Minimum (left) and maximum (right) deformations from PC1 (57.6% of variation). Lower panel: Minimum (left) and maximum (right) deformation from PC2 (15.7% of variation).152

Figure 53. Left: Complete set of semilandmarks (hollow circles) acquired by SoundShape function from advertisement calls of eight *Physalaemus* species. Orange dots represent mean configuration calculated from the complete set of semilandmarks. Right: Simplified grid of semilandmark coordinates using a logarithmic sequence on the time X-axis. Grid encompass 100 cells on the time X-axis and 30 cells on the frequency Y-axis, enabling the acquisition of 3000 semilandmarks from the sound surface. Note: Time coordinates have been magnified 10 times so that time and frequency coordinates will be on the same scale.155

Figure 54. Semilandmarks acquired by eigensound function using advertisement calls of eight *Physalaemus* species. Left: Complete set of three dimensional semilandmarks after Generalized Procrustes Analysis. Right: Consensus shape calculated from the aligned set of three dimensional semilandmarks.157

Figure 55. Eigenvalues as percentage of variance (grey bars) and broken stick model (red line and dots) computed from PCA using Procrustes coordinates. Data acquired by eigensound function using the advertisement call of eight *Physalaemus* species.158

Figure 56. Individual projection of eight *Physalaemus* species onto PC1 and PC2 using Procrustes coordinates acquired by eigensound function. Each point represent one sampled note and convex hulls comprise all notes analyzed for each species. Legend represent colors used for each species.159

Figure 57. Hypothetical semilandmark configurations from minimum (left) and maximum (right) scores of PCA using Procrustes coordinates acquired by eigensound function. Data from the advertisement call of eight *Physalaemus* species. Upper panel:

Hypothetical semilandmark configurations from PC1 (21.1% variance). Lower panel: Minimum (left) and maximum (right) semilandmark configurations from PC2 (18.7% variance).160

Figure 58. Spectrogram (above) and oscillogram (bellow) of a representative note from the advertisement call of *Physalaemus deimaticus* recorded at its type locality, Serra do Cipó, Morro do Pilar Municipality, Minas Gerais State, Brazil.....163

Figure 59. Semilandmarks acquired by `SoundShape` function using advertisement calls of *Physalaemus deimaticus* from three populations. Hollow circles represent complete set of SMs after Generalized Procrustes Analysis using minimum Procrustes distance. Orange dots represent consensus shape calculated from Procrustes coordinates.166

Figure 60. Semilandmarks acquired by `eigensound` function using advertisement calls of *Physalaemus deimaticus* from three populations. Left: Complete set of three dimensional semilandmarks after Generalized Procrustes Analysis. Right: Consensus shape calculated from the aligned set of three dimensional semilandmarks.167

Figure 61. Eigenvalues as percentage of variance (grey bars) and broken stick model (red line and dots) computed from PCA using Procrustes coordinates. Left: Data acquired by `SoundShape` function. Right: Data acquired by `eigensound` function.168

Figure 62. Individual projection of *Physalaemus deimaticus* individuals onto PC1 and PC2 axes using Procrustes coordinates acquired by `SoundShape` function. Each point represent one sampled note and convex hulls comprise all notes analyzed for each population. Legend represent colors used for each individual and localities encompassed by convex hulls. Ind = individual. Loc = locality.....170

Figure 63. Semilandmarks (dots) and deformation grids in relation to consensus configuration (i.e. PC score = 0). Grids calculated using Procrustes coordinates acquired by `SoundShape` function. Data from the advertisement call of three populations of *Physalaemus deimaticus*. Upper panel: Minimum (left) and maximum (right) deformations from PC1 (39.7% variation). Lower panel: Minimum (left) and maximum (right) deformation from PC2 (24.6% variation).....171

Figure 64. Individual projection of *Physalaemus deimaticus* individuals onto PC1 and PC2 axes using Procrustes coordinates acquired by `eigensound` function. Each point represent one sampled note and convex hulls comprise all notes analyzed for each

population. Legend represent colors used for each individual and localities encompassed by convex hulls. Ind = individual. Loc = locality.....172

Figure 65. Hypothetical semilandmark configurations from minimum (left) and maximum (right) scores of PCA using Procrustes coordinates acquired by `eigenstout` function. Data from the advertisement call of three populations of *Physalaemus deimaticus*. Upper panel: Minimum (left) and maximum (right) semilandmark configurations from PC1 (24.6% variation). Lower panel: Minimum (left) and maximum (right) semilandmark configurations from PC2 (17.6% variation).174

LISTA DE TABELAS

APÊNDICE 2

Table 1. Arguments to be used by <code>frequencies</code> function.	52
Table 2. Arguments to be used by <code>SoundShape</code> function.	69
Table 3. Arguments to be used by <code>eigensound</code> function.	104
Table 4. Arguments to be used by broken stick model <code>bsm</code> function.	125
Table 5. Arguments to be used by <code>plot.pca</code> function.	129
Table 6. Arguments to be used by <code>PCshapes</code> function.	135

LISTA DE ABREVIACES

2D – Two dimensions; two dimensional

3D – Three dimensions; three dimensional

CBUFMG – Coleo Bioacstica da Universidade Federal de Minas Gerais

CRA – curve of relative amplitude

CVA – Canonical Variates Analysis

FFT – Fast Fourier Transformation

FNJV – Fonoteca Neotropical Jacques Vielliard

GM – Geometric Morphometrics

GPA – Generalized Procrustes Analysis

LM – Landmark; landmarks

log – logarithmic

PC – Principal Components

PCA – Principal Component Analysis

SM – Semilandmark; semilandmarks

RESUMO

ROCHA, Pedro Carvalho, M.Sc., Universidade Federal de Viçosa, abril de 2018.
A Forma do Som: Um olhar aprofundado sobre a representação gráfica de ondas sonoras. Orientador: Pedro Seyferth Ribeiro Romano. Coorientadores: Renato Neves Feio e Paulo Christiano de Anchieta Garcia.

Em geral, estudos bioacústicos ainda seguem os mesmos protocolos implementados na primeira metade do século XX. Além disso, a maior parte dos parâmetros acústicos tradicionais é variável, estando sujeitos à influência de fatores ambientais e à seleção sexual. Em vista disso, métodos de análise bioacústica precisam ser aprimorados, melhorando a acurácia e precisão em distinguir entre unidades biológicas com cantos semelhantes. No presente trabalho, exploramos a análise previamente descrita *eigensound*, e descrevemos uma nova: *SoundShape*. As duas abordagens contam com a aquisição de pontos cartesianos (*semilandmarks*) a partir de arquivos *wave*, traduzindo, portanto, os dados de um sonograma em configurações de forma (i.e. *shape*) que podem ser avaliadas por ferramentas advindas da Morfometria Geométrica. Ambos os métodos foram codificados dentro da plataforma R, com a descrição de duas novas funções *SoundShape* e *eigensound*. Além disso, a nova função *eigensound* inclui algumas modificações em relação à análise original, de forma a compreender uma maior variedade de sons animais. As funções foram implementadas em dois estudos empíricos utilizando cantos de anúncio de anfíbios anuros pertencentes a várias espécies do gênero *Physalaemus*. O plot de ordenação (PCA) das duas abordagens resultou na estruturação consistente de grupos biológicos (i.e. espécies e populações). *SoundShape* resultou em um maior poder estatístico, enquanto que *eigensound* foi mais sensível à especificidade do canto de cada espécie. *SoundShape* é, portanto, um método eficiente de ordenação de espécies com base nas similaridades no canto de cada uma, sendo também útil como um protocolo de verificação de erros em análises da forma do som. *Eigensound*, por outro lado, é computacionalmente mais exigente, embora seja mais sensível à estrutura sônica de cada canto e capaz de descrever a assinatura vocal de cada indivíduo. Nosso estudo reitera, portanto, o significado biológico da forma do som e a aplicabilidade da Morfometria Geométrica em estudos bioacústicos.

ABSTRACT

ROCHA, Pedro Carvalho, M.Sc., Universidade Federal de Viçosa, April, 2018.
The Shape of Sound: A deeper look into the graphical representation of sound waves. Adviser: Pedro Seyferth Ribeiro Romano. Co-advisers: Renato Neves Feio and Paulo Christiano de Anchieta Garcia.

Generally, bioacoustical studies still follow the same protocols implemented on the first half of the 20th century. Moreover, most traditional call parameters are variable, subject to the influence of environmental factors and sexual selection. In light of this, methods of bioacoustic analyses need to improve the accuracy and precision in distinguishing between biological units with similar calls. We herein explore the previously described *eigensound* method and describe a new one: *SoundShape*. Both approaches rely on the acquisition of point coordinates (semilandmarks) from wave files, thus translating the spectrogram output into shape configurations that can be assessed by the broad range of tools from Geometric Morphometrics. Both methods were coded into R platform, with the description of two new functions *SoundShape* and *eigensound*. Moreover, the new *eigensound* function include some modifications from the original analysis so that it fits a broader range of animal sounds. *SoundShape* and *eigensound* functions were implemented on two empirical studies dealing with anuran advertisement calls from several *Physalaemus* species. Ordination plot (PCA) from both approaches yielded consistent structuring of biological groups (i.e. species and populations). *SoundShape* resulted in higher statistical power, whereas *eigensound* was more sensitive to the specificity of each species call. *SoundShape* is efficient for species ordination based on overall call similarity and is useful as a protocol for error verification in sound shape analysis. *Eigensound* is computationally more demanding, yet very sensitive to the sonic structure of calls, being able to describe the vocal signature of each individual. Thus, our study reiterates the biological meaning of the shape of sound and the applicability of Geometric Morphometrics approach in Bioacoustics.

ARTIGO CIENTÍFICO – Rocha, P.C. & Romano, P.S.R. The Shape of Sound:
Two new $\mathbb{R}$ functions that translate sound waves into point configurations.
Methods in Ecology and Evolution.

The Shape of Sound: Two new R functions that translate sound waves into point configurations

PEDRO C. ROCHA^{1,2} & PEDRO S.R. ROMANO¹

¹*Programa de Pós-Graduação em Biologia Animal, Laboratório de Paleontologia e Osteologia Comparada, Departamento de Biologia Animal, Universidade Federal de Viçosa. 36570-900, Viçosa, MG, Brazil.*

²*Corresponding author: p.rocha1990@gmail.com*

1 – Abstract

1. Generally, bioacoustical studies still follow the same protocols implemented on the middle 20th century. Moreover, most traditional call parameters are variable, subject to the influence of environmental factors and sexual selection. In light of this, methods of bioacoustic analyses need to improve the accuracy and precision in distinguishing between biological units with similar calls.
2. We herein explore the previously described *eigensound* method and describe a new one: *SoundShape*. Both approaches rely on the acquisition of point coordinates (semilandmarks) from wave files, thus translating the spectrogram output into shape configurations that can be assessed by the broad range of tools from Geometric Morphometrics. Both methods were coded into R platform, with the description of two new functions *SoundShape* and *eigensound*. Moreover, the new *eigensound* function include some modifications from the original analysis so that it fits a broader range of animal sounds.
3. *SoundShape* and *eigensound* functions were implemented on two empirical studies dealing with anuran advertisement calls from several *Physalaemus* species. Ordination plot (PCA) from both approaches yielded consistent structuring of biological groups (i.e. species and populations). *SoundShape* resulted in higher statistical power, whereas *eigensound* was more sensitive to the specificity of each species call.
4. *SoundShape* is efficient for species ordination based on overall call similarity and is useful as a protocol for error verification in sound shape analysis. *eigensound* is computationally more demanding, yet very sensitive to the sonic

structure of calls, being able to describe the vocal signature from each individual. Thus, our study reiterates the biological meaning of the shape of sound and the applicability of Geometric Morphometrics approach in Bioacoustics.

2 – Keywords

Bioacoustics, eigensound, Geometric Morphometrics, geomorph, R platform, seewave, semilandmark configuration, SoundShape

3 – Introduction

Bioacoustics is a diverse field focused on animal communication, behavior and the physics of sound production (Fletcher, 2007). Recently, there has been a growing interest for the potential usage of bioacoustics in conservation (Laiolo, 2010), with initiatives towards its application as a tool for biodiversity assessment and monitoring (Pijanowski et al., 2011; Sueur & Farina, 2015), including the development of new methods for species identification (Potamitis, 2014; Towsey, Wimmer, Williamson, & Roe, 2014; Ulloa, Aubin, Llusia, Bouveyron, & Sueur, 2018; Xie, Towsey, Zhang, & Roe, 2018). However the lack of accuracy is still problematic (Sueur & Farina, 2015).

Even though Bioacoustics gained visibility through the ‘integrative taxonomy’ approach (Padial, Miralles, De la Riva, & Vences, 2010; Yeates et al., 2011), call descriptions still follow the same protocols implemented on the middle twentieth century (Gerhardt & Huber, 2002; Köhler et al., 2017). Despite the practicality and objectiveness of traditional bioacoustical descriptions, these protocols focus on a few call parameters, which are mostly correlated (e.g. call duration and number of pulses per call). Not surprisingly, some studies on call diversification argue that acoustic signals are not reliable, and should be carefully considered for systematic purposes (Cannatella et al., 1998; Tobias, Evans, & Kelley, 2011; Robillard & Desutter-Grandcolas, 2011; Toledo, Llusia, Vieira, Corbo, & Márquez, 2015).

Phylogenetic signal in acoustic traits have been found for multiple groups of animals (McCracken & Sheldon, 1997; Price & Lanyon, 2002; Päckert, Martens, Kosuch, Nazarenko, & Veith, 2003; Percy, Taylor, & Kennedy, 2006; Robillard, Höbel, & Gerhardt, 2006; Cap, Deleporte, Joachim, & Reby, 2008; Erdtmann & Amézquita, 2009; Goicoechea, De La Riva, & Padial, 2010). These observations generally agree that acoustic traits constrained by anatomical structures are more conservative than those

under direct influence of behavior. Therefore, it is pivotal that the selection of acoustic units account for the mechanism underlying sound production (Robillard et al., 2006; Cap et al., 2008), thus assuring the homologous comparison between units. In addition, signals are often subject to sexual selection and rapid evolution (Cocroft & Ryan, 1995; Price & Lanyon, 2002; Robillard & Desutter-Grandcolas, 2011), therefore resulting in misleading conclusions about sound evolution (e.g. Tobias, Gamarra-Toledo, García-Olaechea, Pulgarín, & Seddon, 2011; Toledo et al., 2015).

Different acoustic traits are under distinct selective pressures (Gerhardt, 1991; Gerhardt & Huber, 2002). Those that are generally constrained by the calling apparatus are frequently responsible for species recognition (e.g. dominant frequency), and thus conserved within each species (i.e. static parameters) (Gerhardt, 1991). Other traits, however, are variable (e.g. temporal traits), subject to the influence of environmental factors (e.g. temperature), and though to encompass information on mate quality and caller enthusiasm (i.e. dynamic parameters), therefore their use for taxonomic purposes must be done with care (Köhler et al., 2017). On the other hand, anuran species have served as model for several studies on animal sounds (e.g. Narins & Capranica, 1978; Ryan & Rand, 1990; Gerhardt, 1991; Gerhardt, Martínez-Rivera, Schwartz, Marshall, & Murphy, 2007). This is probably because they exhibit highly stereotyped and simple advertisement calls that are species specific and heritable (Köhler et al., 2017), thus being broadly applicable on ethological or methodological studies (Narins & Capranica, 1978; Ryan & Rand, 1990; H Carl Gerhardt, 1991) as it is for taxonomic identification (Blair, 1958; Martof, 1961; Köhler et al., 2017).

To our knowledge, the so called *eigensound* approach (MacLeod, Krieger, & Jones, 2013) is the sole method developed for taxonomy-based bioacoustics using a set of Cartesian coordinates acquired from spectrogram outputs. This method consists on applying a sampling grid over the 3D representation of sound (i.e. spectrogram data) and then translate the spectrogram into a dataset that can be analyzed similarly to coordinate sets used in Geometric Morphometrics. By doing so, MacLeod et al. (2013) successfully managed to discriminate between bat species previously indistinguishable using standard protocols of bioacoustic analysis (e.g. Walters et al., 2012).

Therefore, in light of the current trend for enhanced methods of acoustic analysis – aimed at a higher accuracy in distinguishing species with similar calls – here we present some improvements on *eigensound* approach and describe a new method, *SoundShape*. The new approach – which consists in the acquisition of two dimensional point

coordinates from a spectrogram output – is computationally faster and result in a higher statistical power in relation to *eigensound*. Both methods were coded into R platform (R Core Team, 2017), enabling anyone to reproduce and implement both methods. In addition, we performed two empirical studies in order to implement and compare *eigensound* and *SoundShape*.

4 – Materials and Methods

We based all analyses on empirical data from the calls of 30 species of *Physalaemus* Fitzinger (Amphibia, Anura, Leptodactylidae), a genus currently composed of 48 recognized species (Frost, 2018) classified into two major groups: the *P. cuvieri* and the *P. signifer* clades (Lourenço et al., 2015) (Fig. 1). Each species had one to four individuals sampled according to call availability from bioacoustical collections (see Appendix 1) and each individual had five stereotyped calls selected. We employed the advertisement call as our sound unit because it is the most common type of vocalization in frogs (Toledo et al., 2014), is species specific, and its taxonomic application has long been demonstrated (Blair, 1958; Martof, 1961; Toledo, et al., 2015; Köhler et al., 2017).

Despite advertisement calls being the most comparable sound unit among frogs, it is not always straightforward to define them as homologous. Some anuran species present simple advertisement calls composed of a single acoustic signal (i.e. call = note), whereas other species emit complex calls composed by several units in a series (i.e. call = note series) (Köhler et al., 2017). In order to assert the homology between calls of different species, we adopt herein a physiological definition of ‘note’, in which a note is defined as the total amount of sound energy produced by a single airflow cycle between the lungs, larynx and the vocalsacs (McLister, Stevens, & Bogart, 1995; Robillard et al., 2006). Note definition was not always easy though, given that this approach require an observer to watch the frog during calling activity. Thus, whenever the physiological concept became too arbitrary, we followed the note definition from Köhler et al.(2017), which is focused on the comparable sound units among species.

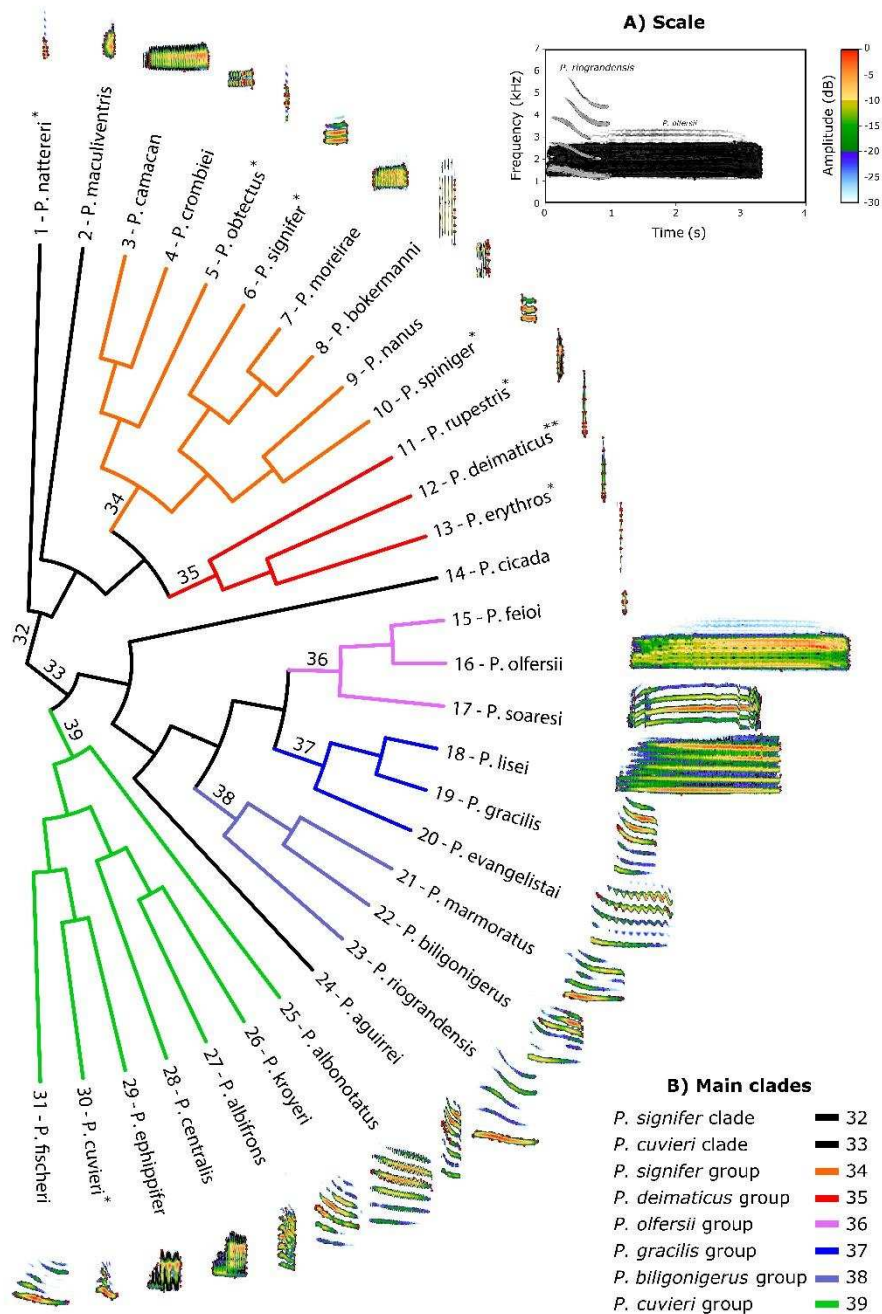


Figure 1: Spectrogram outputs generated by SoundShape function to represent stereotyped advertisement calls (i.e. sound units) of 30 *Physalaemus* species. Spectrograms were colored above the curve of relative amplitude at -25 dB, with red dots representing two-dimensional semilandmarks for each call. A) Sound window (left) and scale of amplitude colors (right) employed on spectrogram outputs. B) Color legend of major clades as proposed by Lourenço et al. (2015). Species marked with asterisks were employed on intraspecific empirical study and *Physalaemus deimaticus* (**) was employed on interpopulational study. Tree topology adapted from Lourenço et al. (2015).

Although there is no universal concept of a homologous unit of biological sound encompassing the majority of calling organisms, each higher taxon has its own approaches for homologous sound comparison (see Appendix 2). Once the sound units are selected respecting the homology criteria adopted, each unit must be saved as a separate wave file and stored in a single folder from which the analysis will be conducted.

Sounds must be set to begin at time = 0 s and remaining of sound window filled with silence. In addition, the length of the sound windows must be defined *a priori* so that the selected length encompasses all sounds in the study. These steps ensure that variation in each point coordinates is due to energy shifts within the call, and not to changes in their relative positions in the sound window (MacLeod et al., 2013).

Even though this work is based on calls from a single genus of frogs, the methods are described as an open code that can be adapted and applied on any given sound, depending on the objectives one might be interested in. Although we explore the empirical data through a taxonomic perspective, both *SoundShape* and *eigensound* approaches can be applied on virtually any bioacoustical study. Ultimately, in order to make both methods more accessible to the academic community, we created an R markdown file (Allaire et al., 2017) containing detailed descriptions and codes for the methods, results, and images employed on the development of *SoundShape* and *eigensound* functions (Appendix 2).

4.1 – New method: *SoundShape* analysis

SoundShape is a new method that consists in the acquisition of point coordinates (i.e. semilandmarks – SM) from a curve of relative amplitude (CRA) (Fig. 2A–D). The analysis is conducted with the function *SoundShape*, which works within R platform (R Core Team, 2017) and require *seewave* (Sueur, Aubin, & Simonis, 2008) and *tuneR* packages (Ligges, Krey, Mersmann, & Schnackenberg, 2016). A systematic and detailed guide to *SoundShape* is provided in Appendix 2. The function code is available in Appendix 3.

The following steps summarize the acquisition of SM by *SoundShape*:

- *SoundShape* function computes point coordinates from a set of wave files stored in a given folder. Each file is the sound wave (i.e. oscillogram; Fig. 2A) that represents a sound unit from the sample.
- Waves are used to generate two dimensional spectrogram outputs (Fig. 2B), which are, essentially, a series of Fast Fourier Transformations (FFT) (i.e. power spectrum) for each discrete time interval and overlapped in the time axis (Charif, Waack, & Strickman, 2010; Köhler et al., 2017). Since the slightest graphical

change may incur in biased results, all spectrogram parameters must be defined *a priori* (see Appendix 2 for details on spectrogram parameters).

- Given that the caller's enthusiasm and its distance from the recording device can impact the spectrogram output of a sound wave (Gerhardt & Huber, 2002; Charif et al., 2010; Köhler et al., 2017), the homologous comparison between sound shapes has to be made from a curve of relative amplitude (CRA). The curve is acquired in relation to the peak of amplitude in the spectrogram, which is typically set as 0 dB and the relative amplitude given as a negative value. `SoundShape` computes all CRA for a given amplitude value and selects the longest one to represent the shape of each sound unit (Fig. 2C). By default, the CRA is calculated at -25 dB, which was experimentally defined as the amplitude value that encompass the general shape of a call with minimal influence of background noise. However, `SoundShape` user can easily change the desired relative amplitude with the argument `dBlevel`.
- CRA are complex curves that encompass comparable information about the shape of sound and each single contour can be objectively treated as homologous within the study sample. Point coordinates, in turn, are acquired by cross-correlation between frequency values and the CRA. By default, `SoundShape` compute seven frequency values relative to quantiles of 5%, 15%, 30%, 50%, 70%, 85% and 95% energy within the square limits of the CRA. Each frequency value yield two SM in the curve, which account for 14 point coordinates. In addition, the minimum and maximum frequency values within the curve (i.e. 0 and 100% energy quantiles) are also selected, thus encompassing 16 SM numbered anticlockwise from the point of highest energy (i.e. highest frequency value). In our sample, this was the maximum number of points that did not cause a computational error in R (e.g. narrow banded notes resulted in overlapped points). However, one can also control the desired energy quantiles with the argument `EQ` and thus acquire more (or less) SM to describe the CRA.
- Ultimately, `SoundShape` discard the spectrogram output and retain only the two dimensional point coordinates representing the shape of sound (Fig. 2D). These coordinates can be assigned to an R object, thus enabling one to proceed with Geometric Morphometric analysis within R platform (i.e. packages `geomorph`, Adams, Collyer, Kaliontzopoulou, & Sherratt, 2017; and `shapes`, Dryden, 2017)

or exported as a TPS from which analysis can be conducted in numerous softwares of geometric analysis of shape (e.g. TPS series, Rohlf, 2015).

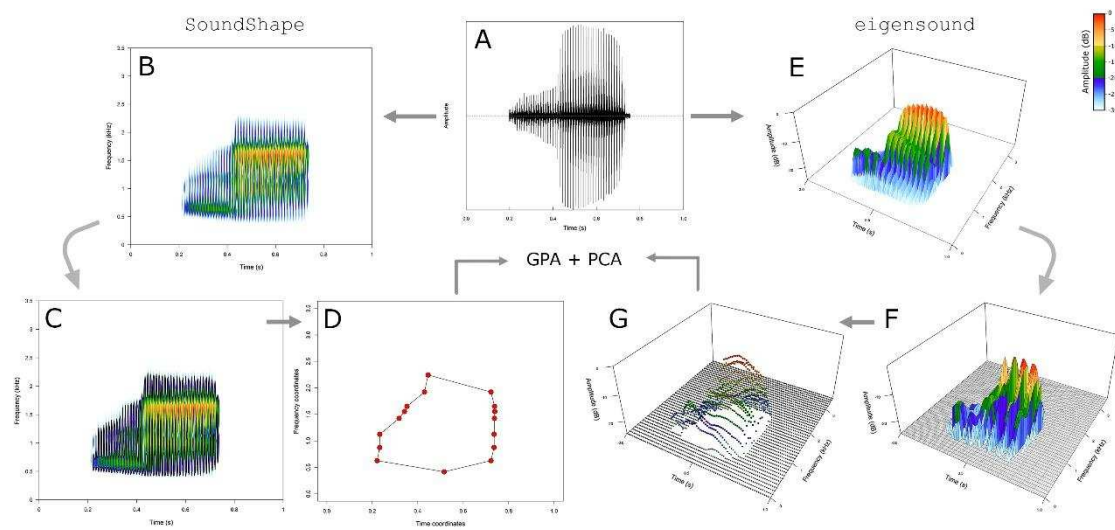


Figure 2: Schematic steps in SoundShape (A–D) and eigensound (A, E–G). A) Both analysis are conducted from sound waves. SoundShape: B) A 2D spectrogram output is created for each wave file on a given folder. C) The longest CRA is computed at -25 dB. D) Point coordinates are acquired by cross correlation between the CRA and frequency values representing energy quantiles. Eigensound: E) Function computes 3D spectrograms for each wave file. F) A sampling grid is superimposed onto the sound surface, resulting in a simplified spectrogram. G) The sampling grid is used to acquire 3D coordinates that represent the shape of sound. 2D and 3D sets of coordinates are subject to Generalized Procrustes Analysis (GPA) and then Principal Components Analysis (PCA) subsequently.

In addition to acquiring 2D coordinates from the spectrogram output of a wave file, SoundShape function can also serve as a protocol for error verification in sound shape analysis. By enabling SoundShape to export image files with spectrogram outputs (see Appendix 2 for details), one can easily verify how SMs are acquired along the CRA for each individual and species in the sample. In addition, the spectrogram images are also useful to verify the appropriate placement of sounds at time = 0 s, which still has to be made manually using any standard software of bioacoustical analysis (e.g. Raven Pro; Bioacoustics Research Program, 2014). These are important steps in sound shape analysis, preventing the comparison of shapes altered by sonic artifacts (e.g. background noise) or overlapping sounds from multiple individuals.

4.2 – Addendum to *eigensound* analysis

In the *eigensound* analysis (MacLeod et al., 2013; Fig. 2A, E–G), a simplified sampling grid is applied over the 3D surface of a spectrogram output (Fig. 2E–F). Each node in the grid bear an amplitude value whose time and frequency coordinates (i.e. x and y-axis) are

identical throughout the grid, meaning that only the amplitude values vary. Therefore, each node in the grid is considered a topologically homologous SM (Fig. 2G), in which x and y represent time and frequency coordinates, respectively, and z represent the amplitude values embedded in the spectrogram.

Eigensound was described in detail elsewhere (MacLeod et al., 2013), but the softwares employed for the analysis were not explicitly provided. Therefore, we implemented the method within R platform and provide a detailed description of it in Appendix 2. In addition, we also developed the new function `eigensound` (Appendix 4), which compute 3D semilandmarks from wave files and store their coordinates either as an R object or TPS file.

In addition, we modified *eigensound* so that it better fits a broader range of biological sounds. In the original proposal, *eigensound* was employed on a sample of short duration calls, which only required a short sound window (i.e. 0.1023 s) and a rather small sampling grid (i.e. 30 or 25 cells per side). However, our sample encompassed species whose call reach almost four seconds, which required a much larger sound window (i.e. 4.0 s) and more cells on each axis of the sampling grid (e.g. 100 cells on the x -axis and 80 cells on the y -axis), thus leading to the acquisition of a large number of SM (i.e. variables). The majority of species from our sample featured short calls (i.e. average duration of 0.05 s), which result in a large “blank” space in the sound window and multiple SM whose coordinates were always identical. That imply on the sampling grid prioritizing the few species with long duration calls, whereas the remaining species had remarkably similar SM configurations. As a way around this issue, we employed a logarithmic (log) sequence on the time x -axis, which emphasize the short duration calls whilst also encompassing the long ones. For further functionalities of `eigensound`, see Appendix 2.

4.3 – Geometric Morphometric analysis

Once the sound waves had been translated into sets of 2D or 3D coordinates (Fig. 3A and E), we can proceed with the sound shape analysis employing tools from Geometric Morphometrics (GM). GM can be summarized as the comparative analysis of shape. By definition, shape is all the geometric information from a configuration of points (i.e. landmarks or semilandmarks) that are invariant to changes in translation, rotation, and scale (Bookstein, 1991; Zelditch, Swiderski, Sheets, & Fink, 2012) (Fig. 3).

Landmarks (LM) are often referred as “homologous points” and defined as discrete anatomical loci that can be recognized as the same point in all specimens in the study (Zelditch et al., 2012). However, when dealing with spectrograms, it is not feasible to obtain discrete homologous points that fit the definition of LM. In the *SoundShape* approach, the 2D coordinates are sampled along a curve of relative amplitude that is objectively treated as homologous within the sample study. Whereas in the *eigensound* approach the 3D coordinates are considered topologically homologous by always possessing the same x and y coordinates (MacLeod et al., 2013). Therefore, the points acquired in both methods are considered analogue or equal to semilandmarks (SM), thus the datasets acquired are mathematically and biologically comparable.

Subsequently, we performed a Generalized Procrustes Analysis (GPA) on sound shape data in order to remove the size (i.e. time and amplitude scales) differences. Moreover, GPA removes differences of position and alignment that are also undesirable in point coordinate data (Bookstein, 1991; Zelditch et al., 2012). Even though the 2D and 3D semilandmark configurations from spectrogram outputs already represent the shape of sound, these configurations also encompass unwanted information (e.g. temperature effects), which can be pragmatically thought as noise and removed from the data by applying a GPA on the original sound shape.

The coordinates (Fig. 3) acquired by *SoundShape* and *eigensound* functions are considered as SM herein and in MacLeod et al. (2013), therefore, they should be adjusted accordingly. Particularly in the *SoundShape* approach, the SM aim to represent a CRA, so it is important to incorporate information about the curve by explicitly defining a set of SMs to slide along the curve. In our study, we allowed SM 2 to 8, and 10 to 16 to slide, whereas SM 1 and 9 (i.e. those with minimum and maximum frequency values) were leaved as LM (i.e. GPA without sliding). Given the very large amount of SM acquired by *eigensound* function, it is not biologically nor computationally feasible to define sliding SM in this method.

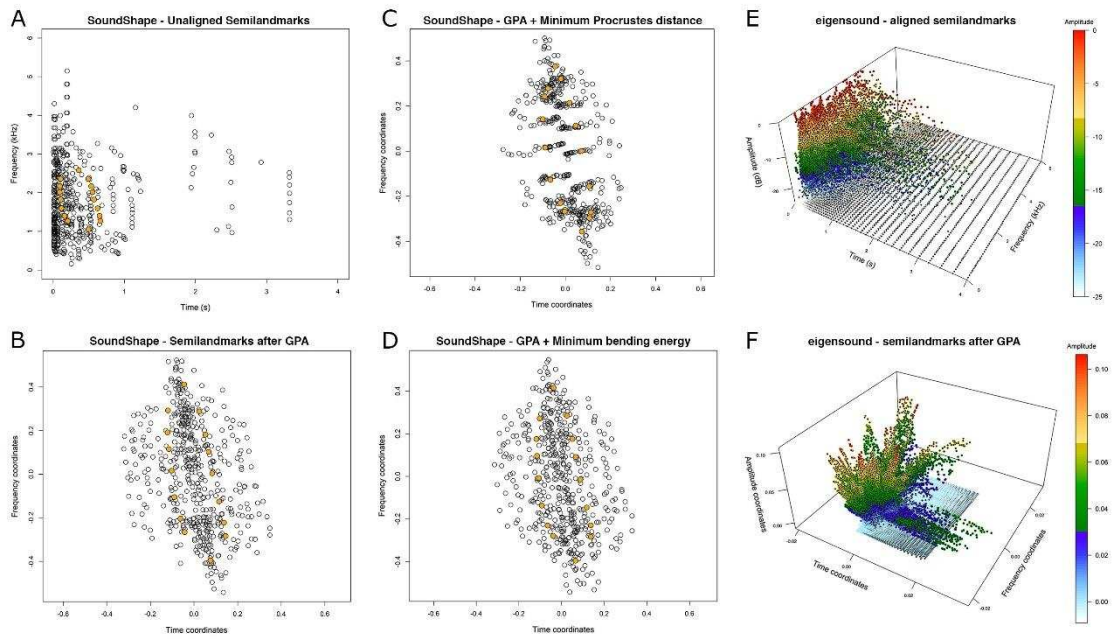


Figure 3: Semilandmark (SM) configurations acquired from the sample of 30 *Physalaemus* species and subject to *SoundShape* (A-D) and *eigensound* (E-F) approaches. A) Complete set of unaligned SM coordinates; B) SM coordinates after Generalized Procrustes Analysis (GPA); C) SM coordinates after GPA using a minimum Procrustes distance D) and a minimum bending energy criterion for sliding SMs; E) Complete set of unaligned SM coordinates; F) SM coordinates after GPA. In A-D, hollow circles represent the complete set (individuals) of 16 SMs and orange circles represent mean coordinates. In E-F, amplitude scale in colors is given.

There are two main approaches to slide SM along an estimated tangent space, which reduce noise in the data and result in better alignments (Bookstein, 1997; Sheets, Covino, Panasiewicz, & Morris, 2006; Zelditch et al., 2012; Rohlf, 2015). The minimum Procrustes distance slides the SM along their tangent directions, minimizing the distance between the adjusted position and the corresponding point in the consensus (Fig. 3C). The minimum bending energy will slide the SM minimizing the energy required for a deformation in relation to the consensus shape (Bookstein, 1997; Zelditch et al., 2012; Rohlf, 2015; Adams et al., 2017) (Fig. 3D). Based on the visual inspection of figures 3B–D, it is clear that the minimum Procrustes distance yielded the best alignment of SM; therefore, we recommend this criterion whenever one is running a GPA on coordinates from *SoundShape*.

The *eigensound* approach, as originally proposed by MacLeod et al. (2013), does not feature a GPA once the SM are acquired (Fig. 3E). However, in our analysis, we conducted GPA on the three dimensional SM coordinates, which reduced some of the variance embedded as noise in the dataset. Once the aligned point coordinates are acquired (Fig. 3F), we proceeded with a Principal Component Analysis (PCA), which is the main ordination method typically employed in GM analysis (Zelditch et al., 2012).

Although a PCA does not represent a test of hypothesis *per se*, the analysis enables ordination plots that encompass the majority of variation embedded in the dataset (Zelditch et al., 2012). However, it is desirable to employ a ‘stopping rule’ to select which PCs should be retained as nontrivial and interpretable and which ones should be ignored (Jackson, 1993). We opted for broken stick models based on eigenvalues (Frontier, 1976), which is a recommended approach for selecting representative PCs (Jackson, 1993). The method consists in the visual interpretation of relative variance from principal components (i.e. eigenvalues or inertia) in relation to the variance expected from a data randomly distributed. The PCs retained by the broken stick criteria are the ones with eigenvalues (i.e. variation) higher than that from a random distribution, thus reducing the probability of selecting PCs that represent random data.

Ultimately, we can also visualize the deformation grids (thin plates splines) as representations of how shape change along the axis of a PCA in relation to consensus configuration (Zelditch et al. 2012). Unfortunately, it is computationally very demanding to produce three dimensional deformation grids due to the large amount of data gathered by `eigensound` function. Therefore, we only provide hypothetical SM configurations from nontrivial PCs of the *eigensound* approach. These graphical outputs enhance the visualization and understanding of ordination plots, thus improving discussion on how sound shape changes along a sample (Zelditch et al., 2012).

Although GM is a broad and diverse discipline that encompasses several additional methods for visualizing, measuring, and even testing hypothesis based on SM configurations, it is not the scope of this work to explore extensively the GM possibilities. Therefore, we will proceed with a couple of studies on empirical data to illustrate how both *SoundShape* and *eigensound* methods perform on real data.

5 – Empirical studies: Results and discussion

5.1 – Sound shape variation between eight *Physalaemus* species

In the first empirical study employing *SoundShape* and *eigensound* functions, we verified how these methods perform in an interspecific scale. To do so, we selected advertisement calls from eight of the 30 *Physalaemus* species employed as empirical data thus far (see Appendix 1 for complete list of recordings; Appendix 2 for detailed

description of the methods and codes employed in the empirical studies; Fig. 1 for phylogenetic relations between species).

Selection encompassed seven species belonging to the *P. signifer* clade (*sensu* Lourenço et al., 2015; Fig. 1), which are either not closely related, but sharing similar spectrogram outputs (i.e. *P. nattereri* and *P. obtectus*; *P. signifer* and *P. spiniger*), or belong to a monophyletic species group that share similar calls (i.e. *P. deimaticus* group, including *P. deimaticus*, *P. erythros* and *P. rupestris*). In addition, we also included a species from the *P. cuvieri* clade (i.e. *P. cuvieri*) with a distinct call. Therefore, this empirical study measures the potential application of both `SoundShape` and `eigensound` functions in structuring species with clearly different spectrogram outputs (e.g. *P. cuvieri* and *P. rupestris*) or remarkably similar calls (e.g. *P. deimaticus* and *P. erythros*) (see Fig. 1 and Appendix 2 for spectrogram outputs).

Aside from *Physalaemus obtectus* (single sampled individual) and *P. erythros* (two sampled individuals) every other species had three individuals sampled according to call availability from bioacoustical collections (Appendix 1). Each individual, in turn, had five stereotyped notes selected according to optimal signal to noise ratio and no overlap with background individuals. Therefore, the interspecific study encompassed 105 notes from 21 individuals belonging to eight *Physalaemus* species. Each note was stored as a separate wave file and placed at time = 0, with remaining of sound window completed with silence. Sound windows were set to range from 0 to 0.5 s and from 0 to 5 kHz, thus encompassing long and short notes within the sample. Spectrogram parameters were set with window length = 512, sampling rate = 44.1 kHz and 80% overlap.

`SoundShape analysis` – Function is first run as protocol for error verification, which assures the correct placement of sounds at time = 0 s, and the appropriate acquisition of SM from the CRA. In addition, this step was useful in defining the length of the sound window and the relative amplitude value (i.e. -25 dB) from which the CRA and the SMs is computed. Once these verifications were concluded, *SoundShape* analysis was run with the default settings, which enable the acquisition of 16 point coordinates from each wave file.

The acquired set of SM coordinates was aligned through a GPA and a minimum Procrustes distance sliding criteria as abovementioned. The new set of aligned SMs represents the Procrustes residuals (Fig. 4A) and is used to perform a PCA on sound

shape data. From the PCA results, we generated a broken stick model (Fig. 4B) for selecting only nontrivial and interpretable PCs (Jackson, 1993).

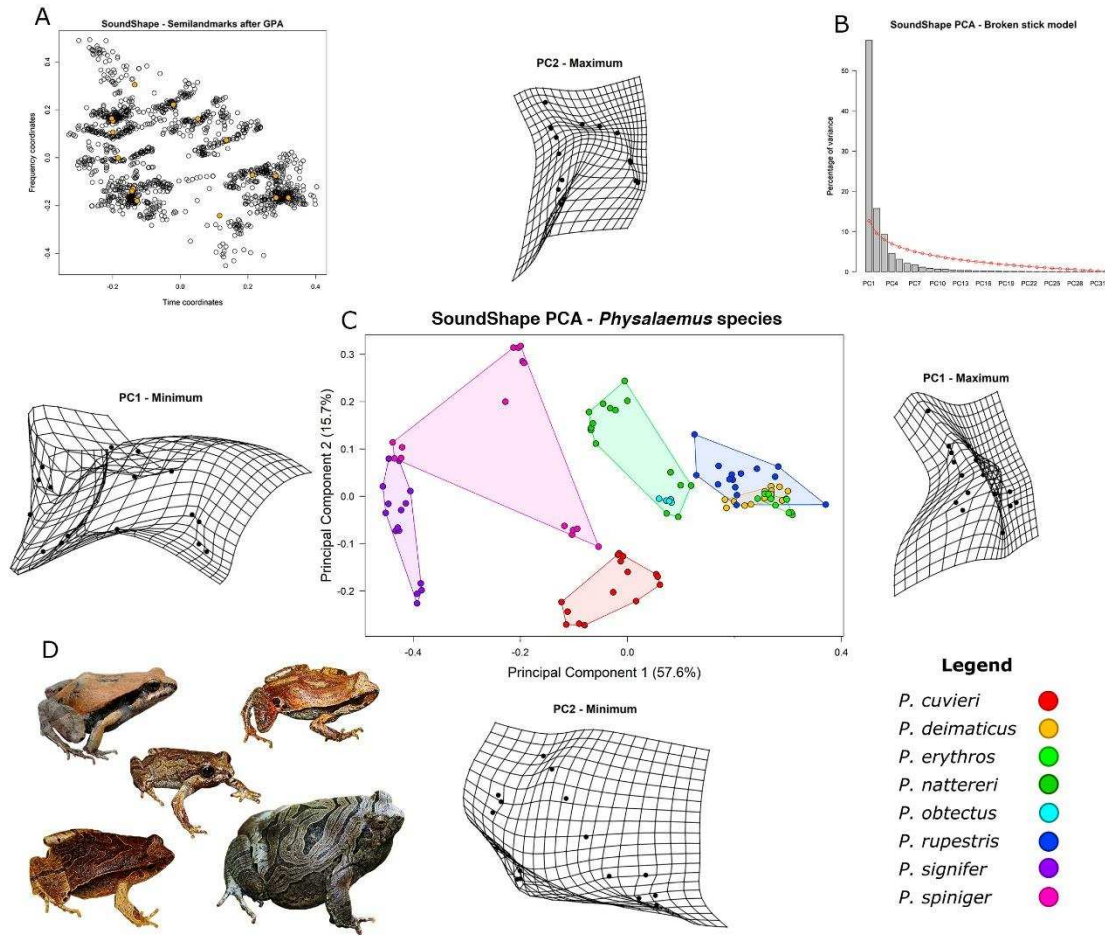


Figure 4: Ordination analysis of SoundShape data. A) Set of SM acquired by SoundShape function from the calls of eight *Physalaemus* species. Hollow circles represent SM after GPA with minimum Procrustes distance criterion for sliding SMs. Orange dots represent consensus shape. B) Eigenvalues as percentage of variance (grey bars) and broken stick model (red line and dots) computed from PCA using Procrustes residuals. C) Individual projection of *Physalaemus* species onto PC1 and PC2. Each point represent one note sampled from individuals and convex hulls comprise all notes analyzed for each species. Semilandmarks (dots) and deformation grids in relation to consensus configuration represent minimum and maximum deformations from PCs and are displayed at the respective edges of ordination plot. Legend represent colors used for each species. D) Unvouchered males of *Physalaemus* species. Top left: *P. cuvieri*; top right: *P. erythros*; center: *P. rupestris*; bottom left: *P. signifer*; bottom right: *P. nattereri*. Photos from F. Leal and L.O. Drummond.

The visual inspection of the broken stick model (Fig. 4B) allow us to retain PC1 (57.6% variance), PC2 (15.7% variance) and PC3 (9.3% variance) as nontrivial. We proceed with the ordination plot of PC1 and PC2, in which both axes encompass 73.3% of the whole variance in our dataset (Fig.4C). Deformation grids relative to minimum and maximum scores of Principal Components 1 and 2 are also displayed.

The first two PCs from SoundShape analysis yielded clear structure among species of *Physalaemus* (Fig. 4C), showing that part of this empirical data is clearly different in shape, which was expected and we interpreted as evidence that SoundShape is a good method for measuring differences between distinct calls. However the species from *Physalaemus deimaticus* group overlapped entirely in the ordination of PC1 and PC2 (although PC3 is not displayed here, it provided similar structuring), which indicate that the *P. deimaticus* group share a common 2D sound shape. A similar result is observed for the pair of *P. nattereri* and *P. obtectus*, which were selected due to the similarity in their spectrogram outputs and scored similar results in the PCA. However, that was not the case for *P. signifer* and *P. spiniger*, with different scores in spite of their similar calls. *Physalaemus cuvieri*, in turn, had no overlap with the other species, a result somewhat expected given the distinctiveness of its call among our sample. Species plotted on the higher PC1 scores were the ones with short, pulsed notes (i.e. *Physalaemus deimaticus* group), whereas those with medium duration and clear harmonic structure scored lower PC1 values (i.e. *P. signifer* and *P. spiniger*). The huge deformation observed in the thin plate splines in minimum PC1 scores (Fig. 4) is due to the presence of harmonic structure in calls of *P. signifer*, *P. spiniger* and *P. cuvieri*.

Eigensound analysis– As described for SoundShape analysis, eigensound yield better results if SoundShape function is first run as protocol for error verification. Subsequently, one must define the sampling grid to be applied over the 3D sound surface and choose between a linear or a log scale on the time x-axis. The disposition of SoundShape unaligned semilandmarks along the sound window (e.g. Fig. 3A) can be useful at this point, as it helps visualizing how SMs are scattered along the sound window. In this empirical study, most SMs were concentrated between 0 and 0.1 s (see Appendix 2 for details), therefore we opted for a log scale with 100 cells on the x-axis and 30 cells on the frequency y-axis, emphasizing the predominance of short duration notes. This grid enabled the acquisition of 3000 SM from the 3D surface of sound, a 75% reduction in data dimensionality in relation to the original grid of amplitude values (i.e. 12449 variables in the original grid).

Once eigensound analysis was ran with the aforementioned parameters, the acquired set of 3D coordinates was subjected to a GPA. The new set of aligned SM (Fig. 5A) was used on a PCA subjected to a broken stick model (Fig. 5B). Although the broken stick model retained the first eight Principal Components as nontrivial, PC6, PC7 and PC8 represent less than 5% variance each. In addition, only PC1, PC2 and PC3 represent

more than 15% variance individually, therefore these are the recommended PCs to use in the ordination plot. We proceed with the ordination of PC1 (21.1% variance) and PC2 (18.8% variance) (Fig. 5C). Hypothetical SM configurations relative to minimum and maximum scores of PCs 1 and 2 are displayed at the extreme edges of the ordination plot (Fig. 5C).

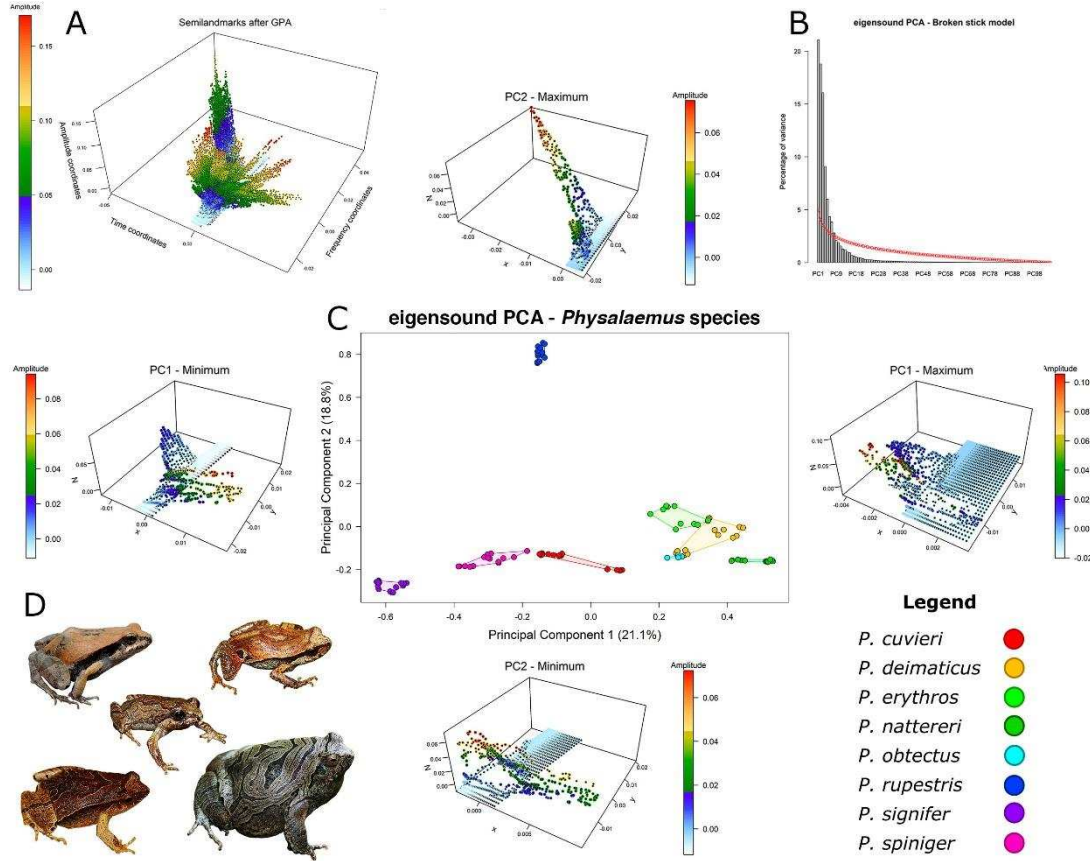


Figure 5: Ordination analysis of *eigensound* data. A) Set of semilandmarks (SM) acquired by *eigensound* function from the calls of eight *Physalaemus* species. SM points are colored in relation to amplitude values. B) Eigenvalues as percentage of variance (grey bars) and broken stick model (red line and dots) computed from PCA using Procrustes residuals. C) Individual projection of *Physalaemus* species onto PC1 and PC2. Each point represent one note sampled from individuals and convex hulls comprise all notes analyzed for each species. Hypothetical SM configurations representing minimum and maximum deformations from PCs are displayed at the respective edges of ordination plot. Legend represent colors used for each species. D) Unvouchered males of *Physalaemus* species. Top left: *P. cuvieri*; top right: *P. erythros*; center: *P. rupestris*; bottom left: *P. signifer*; bottom right: *P. nattereri*. Photos from F. Leal and L.O. Drummond.

Ordination of first two PCs from *eigensound* analysis resulted in remarkable structuring of individuals belonging to the same species (Fig. 5C). Aside from resulting in different scores for species with distinct calls (e.g. *Physalaemus rupestris* and *P. cuvieri*), the *eigensound* method was also sensitive to species with similar advertisement calls (e.g. *P. deimaticus* and *P. erythros*; *P. nattereri* and *P. obtectus*). In

addition, the 3D coordinates scored distinct values according to the acoustic structure of each species call. Those with harmonically structured calls scored low PC2 values, which is in agreement with the results achieved using SoundShape (Fig. 5C).

Despite the SM configurations from PC1 scores being non-intuitive, the scores from species along the PC1 axis indicate that this Component is related to the note duration among species. *Physalaemus obtecus* and *P. nattereri*, for instance, have very similar calls, with short, harmonically structured notes. That granted these species high PC1 scores and low PC2 scores (Fig. 5C). *Physalaemus spiniger* and *P. signifer*, on the other hand, both have harmonically structured calls with medium duration, which granted them low PC1 and PC2 scores.

In addition, the inspection of spectrogram outputs (e.g. Fig. 1; jpeg images generated by SoundShape) yields a reasonable comparison between sound shapes, in which is possible to note that *Physalaemus rupestris* has clearly pulsed structures in the spectrogram output of its call, resulting in the distinct PC scores acquired for this species (Fig. 4C). Therefore, our interspecific study on call shape variation revealed that the eigensound method is sensitive to both the sonic structure and the specificity of advertisement calls in the *Physalaemus* species studied.

5.2 – Sound shape variation between three populations of *Physalaemus deimaticus*

In order to evaluate how SoundShape and eigensound functions perform in the assessment of sound shape changes between populations, we selected calls of *Physalaemus deimaticus* Sazima & Caramaschi 1986 (Anura: Leptodactylidae) from three localities. Three individuals were recorded at each locality, including the species type locality (Serra do Cipó, Morro do Pilar Municipality, Minas Gerais State, Brazil), Diamantina Municipality (Minas Gerais State, Brazil) and Pico do Itambé (Santo Antônio do Itambé Municipality, Minas Gerais State, Brazil; see Appendix 1 for details on recordings). Each individual had five stereotyped notes selected according to optimal signal to noise ratio and no overlap with background individuals, thus encompassing 45 notes stored as separate wave files. Calls were placed at time = 0 s and the remaining sound window completed with silence. Sound windows were set to range from 0 to 0.15 seconds and from 0 to 5 kHz, which encompassed all notes from the sample. Spectrogram parameters were set with window length = 512, sampling rate = 44.1 kHz and 80% overlap.

SoundShape *analysis* – Once the protocol for error verification was performed using SoundShape function (see above), the analysis was conducted with the abovementioned parameters. The acquired set of SM coordinates was aligned through a GPA and a minimum Procrustes distance sliding criteria. The set of aligned SMs represent the Procrustes residuals (Fig. 6A) and is used to perform a PCA on sound shape data. From the PCA results, we generated a broken stick model (Fig. 6B), from which the visual inspection allow us to retain PC1 (39.7% variance), PC2 (24.6% variance) and PC3 (15.6% variance) as nontrivial and interpretable (Jackson, 1993). We proceed with the ordination of PC1 and PC2, in which both axis encompass 64.3% of the whole variance in our dataset (Fig. 6C). Deformation grids relative to minimum and maximum scores of PC1 and PC2 are also displayed.

Regarding the populations from Serra do Cipó and Diamantina, there is an apparent discrepancy in their sound shapes, as structured in PC1 vs PC2 plot (Fig. 6C). This detected difference is consistent with the geographical distance between these two localities (i.e. approximate 110 km). In addition, the overlapping scores between localities is expected and indicate that *Physalaemus deimaticus* has a similar sound shape along its distribution.

Also worth noting that the five notes analyzed for each individual often scored similar PC values. Although there is some variation among analyzed notes – especially in individuals from Diamantina – our results demonstrate that each individual of *Physalaemus deimaticus* has its own sound shape, which vary mainly around the PC1 axis (e.g. individuals “diamantina1”, “diamantina3”, “itambe1” and “itambe3”). The PC1axis, in turn, is responsible for variation on the time x-axis (Fig. 6C), which imply that SoundShape likely encompassed variation in each notes duration, and it reflected on the PC scores and deformation grids.

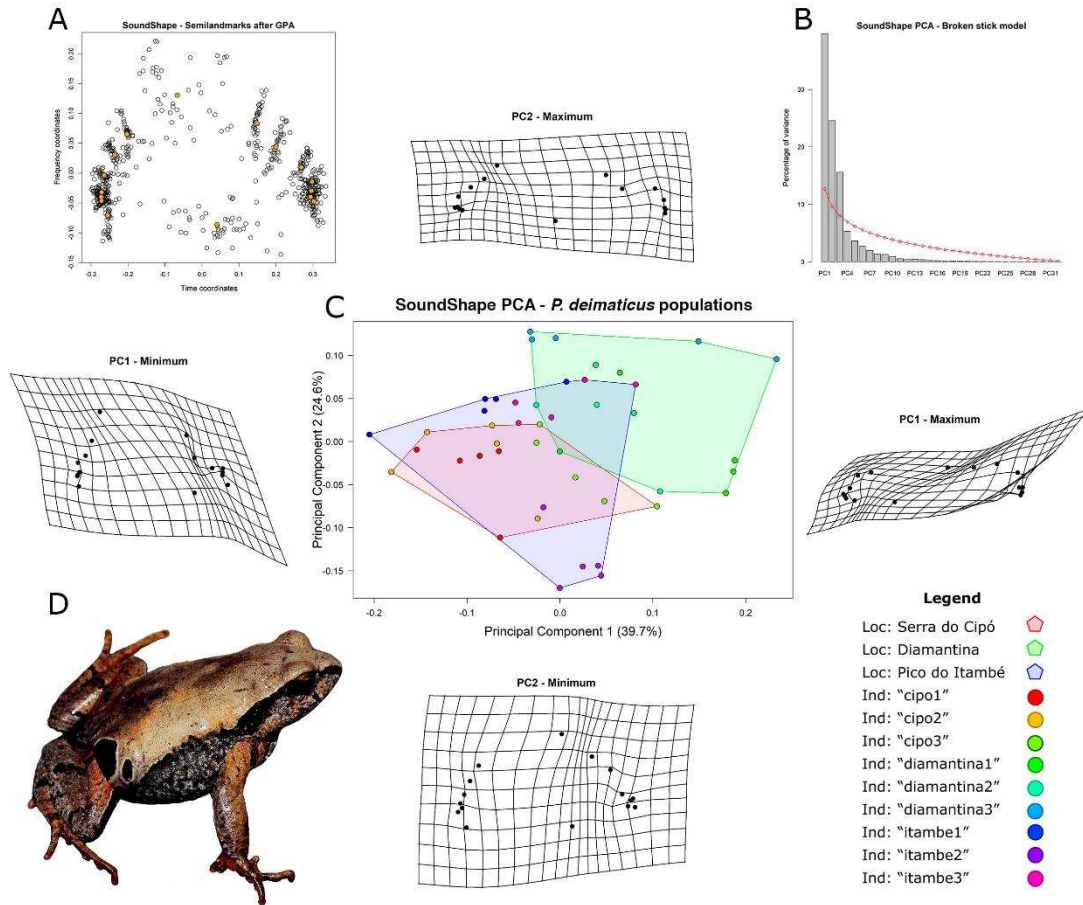


Figure 6: Ordination analysis of SoundShape data. A) Set of SM acquired by SoundShape function using the calls of *Physalaemus deimaticus* from three populations. Hollow circles represent SM after GPA with minimum Procrustes distance criterion for sliding SMs. Orange dots represent consensus shape. B) Eigenvalues as percentage of variance (grey bars) and broken stick model (red line and dots) computed from PCA using Procrustes residuals. C) Individual projection of *P. deimaticus* specimens onto PC1 and PC2. Each point represent one sampled note and convex hulls comprise all notes analyzed for each population. Semilandmarks (dots) and deformation grids in relation to consensus configuration represent minimum and maximum deformations from PCs and are displayed at the respective edges of ordination plot. Legend represent colors used for each individual and localities encompassed by convex hulls. Ind = individual. Loc = locality. D) Unvouchered male of *P. deimaticus* from Serra do Cipó, Morro do Pilar Municipality. Photo from L.O. Drummond.

eigensound analysis – Once the protocol for error verification was performed using SoundShape function (see above), the analysis was conducted using the same parameters employed on interspecific study. However, this study encompass only short notes, so a smaller sound window could be applied (i.e. time-axis = 0 – 0.15 s; frequency-axis = 0 – 5 kHz) and also a smaller sampling grid (i.e. 40 cells on the x-axis and 30 cells on the frequency y-axis) with a linear scale on the time x-axis. The resulting dataset is thus composed of 1200 variables (i.e. semilandmarks), representing a 66.1% reduction in data dimensionality in relation to the original grid of amplitude values (i.e. 3540 variables

in the original grid). The acquired set of three dimensional point coordinates is subjected to a GPA and the new set of aligned SM (Fig. 7A) is used on a PCA subjected to broken stick model (Fig. 7B).

Although the broken stick model retained the first six Principal Components, PC5 and PC6 represent less than 5% variance each. In addition, the variance expressed by those Components is approximately the same as expected for a data randomly distributed, so we do not recommend ordination plots with these PCs. We proceed with the ordination of PC1 (24.6% variance) and PC2 (17.6% variance), in which both axis encompass 42.2% of the whole variance (Fig. 7C). Hypothetical semilandmark configurations relative to minimum and maximum scores of Principal Components 1 and 2 are displayed at the extreme edges of the ordination plot (Fig. 7C).

Although the *eigensound* analysis resulted in less variation expressed by each PC (Fig. 7C), the 3D approach yielded better structuring in the ordination of PCs from the call of *Physalaemus deimaticus* populations. As in the *SoundShape* analysis, the population from Diamantina displayed the greatest variation along both axes, which demonstrate great shape variation in the call from this locality. In addition, the populations from Diamantina and Serra do Cipó displayed the same pattern of accentuated differences in sound shape between populations with great geographical separation, with intermediate scores for the individuals from Pico do Itambé.

Also similar to the result from *SoundShape*, the changes along PC1 axis seems to be related to the relative duration of each note along the sound window, whereas the PC2 changes reflect the bandwidth of each individuals call. That imply on a broad bandwidth (higher PC2 scores) with somewhat variable duration between individuals (low to medium PC1 scores).

Regarding an intrapopulation level, each individual scored similar values in both PCs, which led to structuring of individuals in the ordination plot (Fig. 7C). The exception is the individual “diamantina3”, which had variable notes placed along both PC axis in the 2D and 3D approaches. All other individuals had distinct scores for their calls, placing them along a very narrow space in the ordination plot. This indicate that the *eigensound* method managed to capture the vocal signature from the call of each *Physalaemus deimaticus* individual.

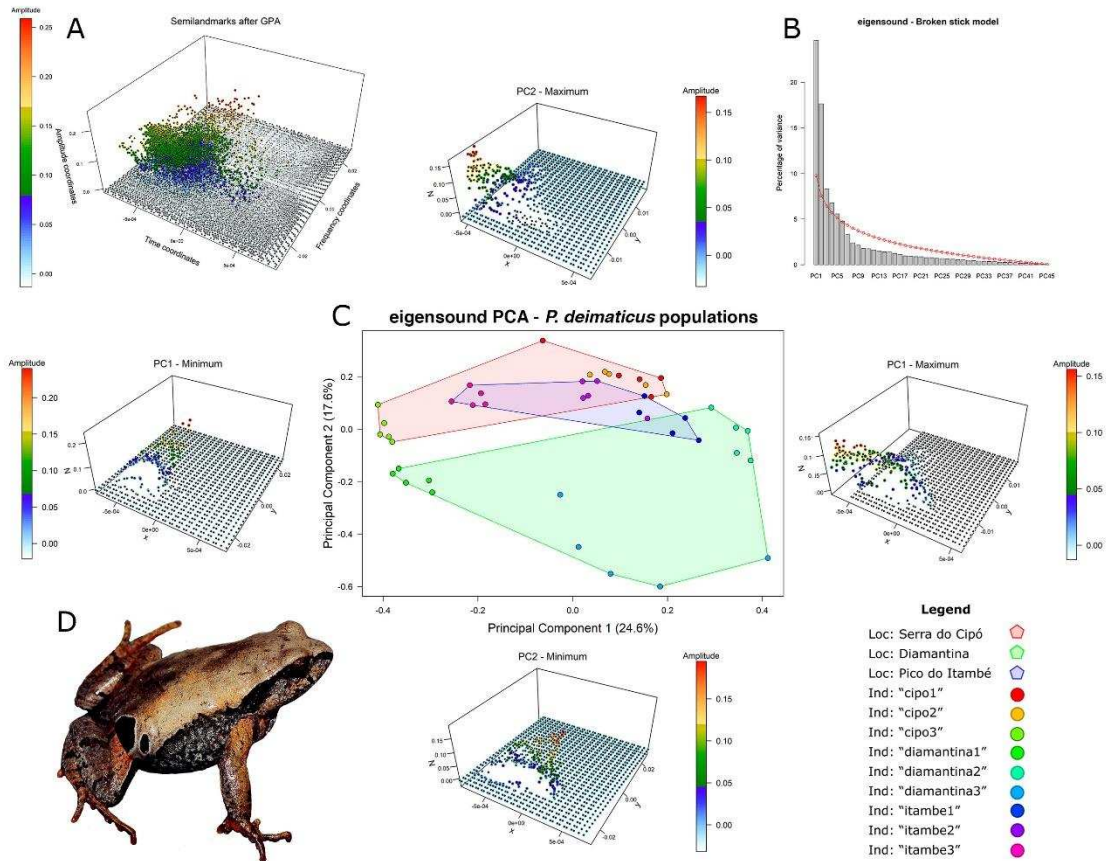


Figure 7: Ordination analysis of *eigensound* data. A) Set of semilandmarks (SM) acquired by *eigensound* function using the calls of *Physalaemus deimaticus* from three populations. SM points are colored in relation to amplitude values. B) Eigenvalues as percentage of variance (grey bars) and broken stick model (red line and dots) computed from PCA using Procrustes coordinates. C) Individual projection of *P. deimaticus* specimens onto PC1 and PC2. Each point represent one sampled note and convex hulls comprise all notes analyzed for each population. Hypothetical SM configurations representing minimum and maximum deformations from PCs are displayed at the respective edges of ordination plot. Legend represent colors used for each individual and localities encompassed by convex hulls. Ind = individual. Loc = locality. D) Unvouchered male of *P. deimaticus* from Serra do Cipó, Morro do Pilar Municipality. Photo from L.O. Drummond

6 – Conclusions

Both SoundShape and *eigensound* methods yielded clear structuring of the calls from *Physalaemus* species. SoundShape function simplify spectrogram outputs into SM described by 2D coordinates, which imply in a computationally less demanding method that result in higher statistical power (higher PCs eigenvalues due to the lower number of variables) in relation to the *eigensound* approach (MacLeod et al., 2013). However, the simplicity of SoundShape is also its main constraint, as it apparently miss some important features of each species calls, thus resulting in ordinations of Principal

Components with groups of species that share similar calls rather than a group for each species. Still, *SoundShape* analysis enables the calculation of deformation grids in relation to the consensus shape (i.e. thin plates splines), which improve discussion on sound shape variation along the sample and provide more reliable visualization of sound structure variation.

In general, *SoundShape* can be thought as a good method for ordination of species based on sound shape similarity. We recommend its usage when one is looking for sonic similarities along a sample of organisms (e.g. shape comparison between monophyletic groups). In addition, *SoundShape* function served as a useful protocol for error verification in sound shape analysis, preventing errors from overlapping individuals or background noise in both *SoundShape* and *eigensound* approaches. The *eigensound* analysis, in turn, is very sensitive. The method successfully encompassed the general structure of each *Physalaemus* call into a few PCs, which resulted in more clear ordination of groups in PC plot. Moreover, the function was accurate enough to result in 3D coordinates that describe the timbre of each individuals call. Yet this high sensitivity is computationally very demanding, which prevents the creation of deformation grids for enhanced discussion of results from PCA. Moreover, *SoundShape* and *eigensound* analysis were both sensitive to the sonic structure of calls, resulting in distinct PC scores according to the presence of harmonic or pulsed structures.

Ultimately, the high sensitivity of *eigensound* analysis encompass a lot of variation from the calls, which remain as noise even after GPA. This variation spreads through several PCs implying on lower statistical power in relation to *SoundShape*. Consequently, the first few PCs are only able to describe part of the variation from the dataset, meaning that several ordination plots would be necessary to comprehend how the sound shape change along a sample. This is not ideal in multivariate analysis and might result in misinterpretation of results.

Despite these constraints, *SoundShape* and *eigensound* analysis stand as promising methods of bioacoustic analysis. *SoundShape* has the advantage of being faster, simpler, and very useful as a protocol for error verification to apply in the exploratory phases of both methods. *Eigensound*, in turn, is a very accurate and sensitive method, being able to decompose the 3D spectrogram output into point coordinates that are unique to each species – and to each individual timbre. To that extent, our empirical

study confirm the conclusion of MacLeod et al. (2013), in which the specificity of each species call is successfully decomposed by the eigensound approach. Although it is still early to draw conclusions on sound shape, it seems clear that there is a biological meaning on the shape of sound, which can be measured by either `SoundShape` or `eigensound` approaches. Future studies might focus on applying distance methods to measure similarities among groups and individual sonograms.

7 – Acknowledgements

We thank Jérôme Sueur for helping with codes employed on `SoundShape`, and Norman MacLeod for insights that helped on the development of `eigensound` function. Carla S. Guimarães helped with images and read early versions of the manuscript. Ministério do Meio Ambiente and Instituto Chico Mendes de Conservação da Biodiversidade granted collection permit 56090-1. Instituto Estadual de Florestas for collection permit 100/2016. Pedro C. Rocha is grateful to Coordenação de Aperfeiçoamento de Pessoal de Nível Superior for MSc. scholarship.

8 – Conflict of interest

The authors declare no conflict of interest.

9 – Authors' contributions

P.C.R. wrote the main text, supplementary materials, R codes and images generated within R platform. P.S.R.R. guided the Geometric Morphometric analysis, helped in the development of `SoundShape` function and revised the text and figures. Both authors contributed to manuscript preparation and revision.

10 – Data Accessibility

The complete list of recordings and collection numbers is displayed on Appendix 1. All codes employed in sound shape analysis are available from Appendix 2.

11 – References

- Adams, D. C., Collyer, M. L., Kaliontzopoulou, A., & Sherratt, E. (2017). Geomorph: Software for geometric morphometric analyses. R package version 3.0.5. Retrieved from <https://cran.r-project.org/package=geomorph>
- Allaire, J. J., Xie, Y., McPherson, J., Luraschi, J., Ushey, K., Atkins, A., ... Chang, W. (2017). rmarkdown: Dynamic documents for R (version 1.8). Retrieved from <https://cran.r-project.org/package=rmarkdown>
- Bioacoustics Research Program. (2014). Raven Pro: Interactive Sound Analysis Software (Version 1.5). Ithaca, NY: The Cornell Lab of Ornithology. Retrieved from <http://www.birds.cornell.edu/raven>
- Blair, W. F. (1958). Mating Call in the Speciation of Anuran Amphibians. *The American Naturalist*, 92(862), 27–51.
- Bookstein, F. L. (1991). *Morphometric tools for landmark data: Geometry and biology*. Cambridge University Press, New York.
- Bookstein, F. L. (1997). Landmark methods for forms without landmarks: morphometrics of group differences in outline shape. *Medical Image Analysis*, 1(3), 225–243.
- Cannatella, D. C., Hillis, D. M., Chippindale, P. T., Weigt, L., Rand, A. S., & Ryan, M. J. (1998). Phylogeny of frogs of the *Physalaemus pustulosus* species group, with an examination of data incongruence. *Systematic Biology*, 47(2), 311–35. doi:10.1080/106351598260932
- Cap, H., Deleporte, P., Joachim, J., & Reby, D. (2008). Male vocal behavior and phylogeny in deer. *Cladistics*, 24(6), 917–931. doi:10.1111/j.1096-0031.2008.00223.x
- Charif, R. A., Waack, A. M., & Strickman, L. M. (2010). *Raven Pro 1.4 User's Manual*. Cornell Lab of Ornithology, Ithaca, NY.
- Cocroft, R. B., & Ryan, M. J. (1995). Patterns of advertisement call evolution in toads and chorus frogs. *Animal Behaviour*, 49(2), 283–303. doi:10.1006/anbe.1995.0043
- Dryden, I. L. (2017). shapes: Statistical Shape Analysis. R package version 1.2.0. Retrieved from <https://cran.r-project.org/package=shapes>

- Erdtmann, L., & Amézquita, A. (2009). Differential evolution of advertisement call traits in dart-poison frogs (anura: Dendrobatidae). *Ethology*, 115(9), 801–811. doi:10.1111/j.1439-0310.2009.01673.x
- Fletcher, N. (2007). Animal Bioacoustics. In *Springer Handbook of Acoustics* (pp. 785–804). New York, NY: Springer, New York. doi:10.1007/978-0-387-30425-0_19
- Frontier, S. (1976). Étude de la décroissance des valeurs propres dans une analyse en composantes principales: Comparaison avec le modèle du bâton brisé. *Journal of Experimental Marine Biology and Ecology*, 25(1), 67–75. doi:10.1016/0022-0981(76)90076-9
- Frost, D. R. (2018). Amphibian Species of the World: an Online Reference. Version 6.0. Retrieved 15 March 2018, from <http://research.amnh.org/herpetology/amphibia/index.html>
- Gerhardt, H. C. (1991). Female mate choice in treefrogs: Static and dynamic properties. *Animal Behaviour*, 42, 615–635.
- Gerhardt, H. C., & Huber, F. (2002). *Acoustic communication in insects and anurans: common problems and diverse solutions*. University of Chicago Press.
- Gerhardt, H. C., Martínez-Rivera, C. C., Schwartz, J. J., Marshall, V. T., & Murphy, C. G. (2007). Preferences based on spectral differences in acoustic signals in four species of treefrogs (Anura: Hylidae). *The Journal of Experimental Biology*, 210(Pt 17), 2990–2998. doi:10.1242/jeb.006312
- Goicoechea, N., De La Riva, I., & Padial, J. M. (2010). Recovering phylogenetic signal from frog mating calls. *Zoologica Scripta*, 39(2), 141–154. doi:10.1111/j.1463-6409.2009.00413.x
- Jackson, D. A. (1993). Stopping rules in Principal Components Analysis: A comparison of heuristical and statistical approaches. *Ecology*, 74(8), 2204–2214.
- Köhler, J., Jansen, M., Rodriguez, A., Kok, P. J. R., Toledo, L. F., Emmrich, M., ... Vences, M. (2017). The use of bioacoustics in anuran taxonomy: theory, terminology, methods and recommendations for best practice. *Zootaxa*, 4251(1), 1–124. doi:10.11646/zootaxa.4251.1.1
- Laiolo, P. (2010). The emerging significance of bioacoustics in animal species conservation. *Biological Conservation*, 143(7), 1635–1645.

doi:10.1016/j.biocon.2010.03.025

- Ligges, U., Krey, S., Mersmann, O., & Schnackenberg, S. (2016). tuneR: Analysis of music. Retrieved from <http://r-forge.r-project.org/projects/tuner/>
- Lourenço, L. B., Targueta, C. P., Baldo, D., Nascimento, J., Garcia, P. C. A., Andrade, G. V, ... Recco-Pimentel, S. M. (2015). Phylogeny of frogs from the genus *Physalaemus* (Anura, Leptodactylidae) inferred from mitochondrial and nuclear gene sequences. *Molecular Phylogenetics and Evolution*, 92, 204–16.
doi:10.1016/j.ympev.2015.06.011
- MacLeod, N., Krieger, J., & Jones, K. E. (2013). Geometric morphometric approaches to acoustic signal analysis in mammalian biology. *Hystrix, the Italian Journal of Mammalogy*, 24(1), 110–125. doi:10.4404/hystrix-24.1-6299
- Martof, B. S. (1961). Vocalization as an isolating mechanism in frogs. *American Midland Naturalist*, 65(1), 118–126. doi:10.2307/2423007
- McCracken, K. G., & Sheldon, F. H. (1997). Avian vocalizations and phylogenetic signal. *Proceedings of the National Academy of Sciences of the United States of America*, 94(8), 3833–3836. doi:10.1073/pnas.94.8.3833
- McLister, J. D., Stevens, E. D., & Bogart, J. P. (1995). Comparative contractile dynamics of calling and locomotor muscles in three hylid frogs. *The Journal of Experimental Biology*, 198, 1527–1538.
- Narins, P. M., & Capranica, R. R. (1978). Communicative significance of two-note call of treefrog *Eleutherodactylus coqui*. *Journal of Comparative Physiology*, 127(1), 1–9.
- Päckert, M., Martens, J., Kosuch, J., Nazarenko, A. A., & Veith, M. (2003). Phylogenetic signal in the song of crests and kinglets (Aves: Regulidae). *Evolution*, 57(3), 616–629. doi:10.1111/j.0014-3820.2003.tb01553.x
- Padial, J. M., Miralles, A., De la Riva, I., & Vences, M. (2010). The integrative future of taxonomy. *Frontiers in Zoology*, 7(1), 1–14. doi:10.1186/1742-9994-7-16
- Percy, D. M., Taylor, G. S., & Kennedy, M. (2006). Psyllid communication: Acoustic diversity, mate recognition and phylogenetic signal. *Invertebrate Systematics*, 20(4), 431–445. doi:10.1071/IS05057
- Pijanowski, B. C., Villanueva-Rivera, L. J., Dumyahn, S. L., Farina, A., Krause, B. L.,

- Napoletano, B. M., ... Pieretti, N. (2011). Soundscape Ecology: The science of sound in the landscape. *BioScience*, 61(3), 203–216. doi:10.1525/bio.2011.61.3.6
- Potamitis, I. (2014). Automatic classification of a taxon-rich community recorded in the wild. *PLoS ONE*, 9(5), 1–11. doi:10.1371/journal.pone.0096936
- Price, J. J., & Lanyon, S. M. (2002). Reconstructing the evolution of complex bird song in the Oropendolas. *Evolution*, 56(7), 1514–1529. doi:10.1554/0014-3820(2002)056[1514:RTEOCB]2.0.CO;2
- R Core Team. (2017). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. Retrieved from <https://www.r-project.org/>
- Robillard, T., & Desutter-Grandcolas, L. (2011). Evolution of calling songs as multicomponent signals in crickets (Orthoptera: Grylloidea: Eneopterinae). *Behaviour*, 148(5), 627–672. doi:10.1163/000579511X572044
- Robillard, T., Höbel, G., & Gerhardt, H. C. (2006). Evolution of advertisement signals in North American hyliid frogs: vocalizations as end-products of calling behavior. *Cladistics*, 22(6), 533–545. doi:10.1111/j.1096-0031.2006.00118.x
- Rohlf, F. J. (2015). The tps series of software. *Hystrix*, 26(1), 9–12. doi:10.4404/hystrix-26.1-11264
- Ryan, M. J., & Rand, A. S. (1990). The sensory basis of sexual selection for complex calls in the tungara frog, *Physalaemus pustulosus* (sexual selection for sensory exploitation). *Evolution*, 44(2), 305–314. doi:10.2307/2409409
- Sheets, H. D., Covino, K. M., Panasiewicz, J. M., & Morris, S. R. (2006). Comparison of geometric morphometric outline methods in the discrimination of age-related differences in feather shape. *Frontiers in Zoology*, 3(15), 1–12. doi:10.1186/1742-9994-3-15
- Sueur, J., Aubin, T., & Simonis, C. (2008). Seewave, a free modular tool for sound analysis and synthesis. *Bioacoustics*, 18(2), 213–226. doi:10.1080/09524622.2008.9753600
- Sueur, J., & Farina, A. (2015). Ecoacoustics: the ecological investigation and interpretation of environmental sound. *Biosemitics*, 8(3), 493–502. doi:10.1007/s12304-015-9248-x

- Tobias, J. A., Gamarra-Toledo, V., García-Olaechea, D., Pulgarín, P. C., & Seddon, N. (2011). Year-round resource defence and the evolution of male and female song in suboscine birds: Social armaments are mutual ornaments. *Journal of Evolutionary Biology*, 24(10), 2118–2138. doi:10.1111/j.1420-9101.2011.02345.x
- Tobias, M., Evans, B. J., & Kelley, D. B. (2011). Evolution of advertisement calls in African clawed frogs. *Behaviour*, 148(4), 519–549. doi:10.1163/000579511X569435
- Toledo, L. F., Llusia, D., Vieira, C. A., Corbo, M., & Márquez, R. (2015). Neither convergence nor divergence in the advertisement call of sympatric congeneric Neotropical treefrogs. *Bioacoustics*, 24(1), 31–47. doi:10.1080/09524622.2014.926831
- Toledo, L. F., Martins, I. A., Bruschi, D. P., Passos, M. A., Alexandre, C., & Haddad, C. F. B. (2015). The anuran calling repertoire in the light of social context. *Acta Ethologica*, 18(2), 87–99. doi:10.1007/s10211-014-0194-4
- Towsey, M., Wimmer, J., Williamson, I., & Roe, P. (2014). The use of acoustic indices to determine avian species richness in audio-recordings of the environment. *Ecological Informatics*, 21(100), 110–119. doi:10.1016/j.ecoinf.2013.11.007
- Ulloa, J. S., Aubin, T., Llusia, D., Bouveyron, C., & Sueur, J. (2018). Estimating animal acoustic diversity in tropical environments using unsupervised multiresolution analysis. *Ecological Indicators*, 90, 346–355. doi:10.1016/j.ecolind.2018.03.026
- Walters, C. L., Freeman, R., Collen, A., Dietz, C., Fenton, M. B., Jones, G., ... Jones, K. E. (2012). A continental-scale tool for acoustic identification of European bats. *Journal of Applied Ecology*, 49(5), 1064–1074.
- Xie, J., Towsey, M., Zhang, J., & Roe, P. (2018). Frog call classification: a survey. *Artificial Intelligence Review*, 49(3), 375–391. doi:10.1007/s10462-016-9529-z
- Yeates, D. K., Seago, A., Nelson, L., Cameron, S. L., Joseph, L., & Trueman, J. W. H. (2011). Integrative taxonomy, or iterative taxonomy? *Systematic Entomology*, 36(2), 209–217. doi:10.1111/j.1365-3113.2010.00558.x
- Zelditch, M. L., Swiderski, D. L., Sheets, H. D., & Fink, W. L. (2012). *Geometric morphometrics for biologists: A primer*. Elsevier (Second Edi). Elsevier, San Diego. doi:10.1016/B978-0-12-386903-6.00001-0

12 – Appendix 1: Recordings of *Physalaemus* employed as empirical data in the development of *SoundShape* and *eigensound* methods.

Collection ID	Species	Country	State	Municipality	Temperature	Date	Coletor
FNJV_0031167	<i>P. aguirrei</i>	Brazil	Espírito Santo	Linhares	21	15/02/1990	Cardoso, A .J.
FNJV_0031776	<i>P. aguirrei</i>	Brazil	Espírito Santo	Linhares	20	03/11/1964	Bokermann, W.C.A.
FNJV_0031943	<i>P. albifrons</i>	Brazil	Bahia	Maracás	20	24/11/1965	Bokermann, W.C.A.
FNJV_0031171	<i>P. albonotatus</i>	Brazil	Mato Grosso	Cáceres	30	12/12/1991	Cardoso, A .J.
FNJV_0031172	<i>P. albonotatus</i>	Brazil	Mato Grosso	Cáceres	28	13/12/1991	Cardoso, A .J.
FNJV_0031176	<i>P. albonotatus</i>	Brazil	Mato Grosso	Cáceres	25	17/12/1992	Cardoso, A .J.
FNJV_0031682	<i>P. biligonigerus</i>	Brazil	Rio Grande do Sul	Osório	22	14/12/1996	Cardoso, A .J.
FNJV_0031182	<i>P. bokermanni</i>	Brazil	São Paulo	Santo André	19	20/11/1981	Cardoso, A .J.
FNJV_0031183	<i>P. bokermanni</i>	Brazil	São Paulo	Santo André	18	20/11/1981	Cardoso, A .J.
FNJV_0031184	<i>P. bokermanni</i>	Brazil	São Paulo	Santo André	18	20/11/1981	Cardoso, A .J.
FNJV_0034021	<i>P. camacan</i>	Brazil	Bahia	Igrapiúna	19.5	03/09/2010	Mira-Mendes, C.V.
FNJV_0031187	<i>P. centralis</i>	Brazil	Santa Catarina	São Bento do Sul	26	20/01/1989	Cardoso, A .J.
FNJV_0031188	<i>P. centralis</i>	Brazil	Tocantins	Formoso do Araguaia	25	09/12/1992	Cardoso, A .J.
FNJV_0031234	<i>P. centralis</i>	Brazil	Mato Grosso	Cáceres	26	18/11/1988	Cardoso, A .J.
FNJV_0031235	<i>P. centralis</i>	Brazil	Mato Grosso	Barra do Bugres	24	21/11/1988	Cardoso, A .J.
FNJV_0031191	<i>P. cicada</i>	Brazil	Paraíba	Patos	23	27/01/1990	Cardoso, A .J.
FNJV_0032255	<i>P. crombiei</i>	Brazil	Espírito Santo	Aracruz	23.5	24/10/1996	Haddad, C. F. B.
CBUFMG_142	<i>P. cuvieri</i>	Brazil	Minas Gerais	Carlos Chagas	20	29/10/2013	Rocha, P. C.
CBUFMG_196	<i>P. cuvieri</i>	Brazil	São Paulo	São José dos Campos	22	24/09/2013	Rocha, P. C.
CBUFMG_240	<i>P. cuvieri</i>	Brazil	Mato Grosso	Chapada dos Guimarães	24	14/12/2013	Rocha, P. C.
CBUFMG_173	<i>P. deimaticus</i>	Brazil	Minas Gerais	Jaboticatubas	19	19/10/2013	Thomassen, H.
CBUFMG_174	<i>P. deimaticus</i>	Brazil	Minas Gerais	Jaboticatubas	18.5	19/10/2013	Thomassen, H.
CBUFMG_181	<i>P. deimaticus</i>	Brazil	Minas Gerais	Jaboticatubas	19.2	19/10/2013	Thomassen, H.
CBUFMG_106	<i>P. deimaticus</i>	Brazil	Minas Gerais	Santo Antônio do Itambé	-	21/11/2012	Rocha, P. C.

Appendix 1 (cont.): Recordings of *Physalaemus*.

Collection ID	Species	Country	State	Municipality	Temperature	Date	Coletor
CBUFMG_881	<i>P. deimaticus</i>	Brazil	Minas Gerais	Diamantina	23	29/12/2016	Guimarães, C.S.
FNJV_0031234	<i>P. centralis</i>	Brazil	Mato Grosso	Cáceres	26	18/11/1988	Cardoso, A .J.
FNJV_0031235	<i>P. centralis</i>	Brazil	Mato Grosso	Barra do Bugres	24	21/11/1988	Cardoso, A .J.
CBUFMG_69	<i>P. erythros</i>	Brazil	Minas Gerais	Diamantina	-	30/10/2011	Leal, F.
CBUFMG_119	<i>P. erythros</i>	Brazil	Minas Gerais	Barão de Cocais	22	25/10/2013	Rocha, P. C.
CBUFMG_121	<i>P. erythros</i>	Brazil	Minas Gerais	Barão de Cocais	22	25/10/2013	Rocha, P. C.
CBUFMG_123	<i>P. erythros</i>	Brazil	Minas Gerais	Barão de Cocais	22	25/10/2013	Rocha, P. C.
CBUFMG_152	<i>P. erythros</i>	Brazil	Minas Gerais	Ouro Preto	20	20/10/2013	Rocha, P. C.
CBUFMG_154	<i>P. erythros</i>	Brazil	Minas Gerais	Ouro Preto	20	20/10/2013	Rocha, P. C.
CBUFMG_867	<i>P. erythros</i>	Brazil	Minas Gerais	Catas Altas	10	14/06/2016	Rocha, P. C.
CBUFMG_871	<i>P. erythros</i>	Brazil	Minas Gerais	Catas Altas	14	17/11/2016	Rocha, P. C.
FNJV_0031631	<i>P. evangelistai</i>	Brazil	Minas Gerais	Santana do Riacho	21	11/01/1996	Cardoso, A .J.
FNJV_0031636	<i>P. evangelistai</i>	Brazil	Minas Gerais	Santana do Riacho	16.5	13/01/1996	Cardoso, A .J.
FNJV_0031900	<i>P. evangelistai</i>	Brazil	Minas Gerais	Serra do Cipó	19	10/02/1964	Bokermann, W.C.A.
PCR_V392	<i>P. feioi</i>	Brazil	Minas Gerais	Viçosa	23	17/01/2017	Rocha, P. C.
FNJV_0031231	<i>P. fischeri</i>	Venezuela	Apure	Mantecal	25	07/06/1990	Cardoso, A .J.
FNJV_0031232	<i>P. fischeri</i>	Venezuela	Apure	Mantecal	23	08/06/1990	Cardoso, A .J.
FNJV_0031609	<i>P. gracilis</i>	Brazil	Rio Grande do Sul	Porto Alegre	19	08/12/1995	Cardoso, A .J.
FNJV_0031800	<i>P. kroyeri</i>	Brazil	Minas Gerais	Teófilo Otoni	24	17/11/1968	Bokermann, W.C.A.
FNJV_0032047	<i>P. kroyeri</i>	Brazil	Bahia	Ilhéus	24	05/08/1972	Bokermann, W.C.A.
FNJV_0031623	<i>P. lisei</i>	Brazil	Rio Grande do Sul	São Francisco de Paula	18	11/12/1995	Cardoso, A .J.
FNJV_0031645	<i>P. lisei</i>	Brazil	Rio Grande do Sul	São Francisco de Paula	-	10/12/1995	Cardoso, A .J.
FNJV_0031815	<i>P. maculiventris</i>	Brazil	São Paulo	Paranapiacaba	20	13/01/1963	Bokermann, W.C.A.

Appendix 1 (cont.): Recordings of *Physalaemus*.

Collection ID	Species	Country	State	Municipality	Temperature	Date	Coletor
FNJV_0033756	<i>P. maculiventris</i>	Brazil	São Paulo	Salesópolis	21.5	08/12/1976	Frogs of Boracéia
FNJV_0004592	<i>P. marmoratus</i>	Brazil	São Paulo	Luis Antônio	-	08/01/1993	Cardoso, A .J.
FNJV_0033755	<i>P. moreirae</i>	Brazil	São Paulo	Salesópolis	21.5	06/12/1976	Frogs of Boracéia
FNJV_0031597	<i>P. nanus</i>	Brazil	Santa Catarina	Rancho Queimado	-	28/11/1995	Cardoso, A .J.
FNJV_0031601	<i>P. nanus</i>	Brazil	Santa Catarina	Urubici	17.5	29/11/1995	Cardoso, A .J.
FNJV_0031252	<i>P. nattereri</i>	Brazil	São Paulo	Botucatu	21.5	03/12/1988	Pomba Jr, J. P.
FNJV_0031253	<i>P. nattereri</i>	Brazil	Mato Grosso	Cáceres	26	18/11/1988	Cardoso, A .J.
FNJV_0031628	<i>P. nattereri</i>	Brazil	São Paulo	Luis Antônio	23.5	08/01/1996	Cardoso, A .J.
FNJV_0032880	<i>P. nattereri</i>	Brazil	Goiás	Piracanjuba	-	19/02/2000	Bastos, R. P.
FNJV_0031774	<i>P. obtectus</i>	Brazil	Espírito Santo	Linhares	19.5	02/11/1964	Bokermann, W.C.A.
FNJV_0031559	<i>P. olfersii</i>	Brazil	Minas Gerais	Viçosa	-	17/12/1994	Cardoso, A .J.
FNJV_0031560	<i>P. olfersii</i>	Brazil	Minas Gerais	Viçosa	-	17/12/1994	Cardoso, A .J.
FNJV_0031263	<i>P. olfersii</i>	Brazil	São Paulo	São José do Barreiro	19	12/02/1987	Cardoso, A .J.
FNJV_0031610	<i>P. riograndensis</i>	Brazil	Rio Grande do Sul	Eldorado do Sul	19	09/12/1995	Cardoso, A .J.
FNJV_0031643	<i>P. riograndensis</i>	Brazil	Rio Grande do Sul	Eldorado do Sul	16	09/12/1995	Cardoso, A .J.
CBUFMG_874	<i>P. rupestris</i>	Brazil	Minas Gerais	Lima Duarte	15.5	18/12/2016	Rocha, P. C.
CBUFMG_875	<i>P. rupestris</i>	Brazil	Minas Gerais	Lima Duarte	15.5	18/12/2016	Rocha, P. C.
CBUFMG_876	<i>P. rupestris</i>	Brazil	Minas Gerais	Lima Duarte	15.5	18/12/2016	Rocha, P. C.
CBUFMG_877	<i>P. rupestris</i>	Brazil	Minas Gerais	Lima Duarte	18	19/12/2016	Rocha, P. C.
CBUFMG_886	<i>P. rupestris</i>	Brazil	Minas Gerais	Rio Preto	17.5	02/12/2016	Rocha, P. C.
CBUFMG_887	<i>P. rupestris</i>	Brazil	Minas Gerais	Rio Preto	17.5	02/12/2016	Rocha, P. C.
CBUFMG_888	<i>P. rupestris</i>	Brazil	Minas Gerais	Rio Preto	17.5	02/12/2016	Rocha, P. C.
FNJV_0031273	<i>P. signifer</i>	Brazil	Rio de Janeiro	Itaguaí	23	06/12/1982	Cardoso, A .J.
FNJV_0031276	<i>P. signifer</i>	Brazil	Rio de Janeiro	Itaguaí	23	09/12/1982	Cardoso, A .J.
FNJV_0031919	<i>P. signifer</i>	Brazil	Rio de Janeiro	Itaguaí	23.5	23/01/1965	Bokermann, W.C.A.

Appendix 1 (cont.): Recordings of *Physalaemus*.

Collection ID	Species	Country	State	Municipality	Temperature	Date	Coletor
FNJV_0031272	<i>P. soaresi</i>	Brazil	Rio de Janeiro	Itaguaí	23	09/12/1982	Cardoso, A .J.
FNJV_0031570	<i>P. spiniger</i>	Brazil	São Paulo	Caraguatatuba	23	15/03/1995	Cardoso, A .J.
FNJV_0031790	<i>P. spiniger</i>	Brazil	São Paulo	Jacupiranga	20	15/08/1964	Bokermann, W.C.A.

The Shape of Sound made step by step

Pedro C. Rocha, Pedro S.R. Romano

April 2018

Contents

First things first	36
Quick note for begginers in R.....	36
Packages required.....	37
A brief introduction	38
<i>SoundShape</i> 2D analysis: Semilandmarks from a curve of relative amplitude	39
Detection of relative amplitude contours.....	39
Acquisition of semilandmarks coordinates to use in Geometric Morphmetric analysis.....	49
SoundShape function: Acquiring 2D semilandmarks from multiple wave file.....	69
Using SoundShape function.....	76
<i>eigensound</i> 3D analysis: Semilandmarks from the three dimensional surface of sound	85
Understanding the matrix of amplitude values.....	85
Reducing data dimensionality in the matrix of amplitude values.....	94
eigensound function: Acquiring 3D semilandmarks from multiple wave files.....	104
Using eigensound function.....	
Geometric Morphometrics applied to bioacoustics	118
Defining the Geometric Morphometric methods of shape analysis in R.....	118
Principal Component Analysis (PCA) of shape.....	126

Concluding remarks.....	143
The shape of sound within <i>Physalaemus</i> (Anura: Leptodactylidae): An empirical study employing SoundShape and eigensound functions	145
The sound shape between <i>Physalaemus</i> species.....	145
Interpopulational variation in the sound shape of <i>Physalaemus deimaticus</i> Sazima & Caramaschi 1986.....	164
Concluding remarks from empirical studies of SoundShape and eigensound methods.....	177
References	178

First things first

The methods described herein are a inextricable part of the paper *The Shape of Sound: Two new R functions that translate sound waves into semilandmarks* (Rocha & Romano *in prep*). Anyone who wishes to publish results using the methods described therein must cite the paper describing it.

Rocha, P. & Romano, P. (*in prep*) The shape of sound: A deeper look into the graphical representation of sound. *Methods in Ecology and Evolution*.

In addition, we would appreciate if you cite this supplementary file when employing the methods described herein.

Rocha, P. & Romano, P. (*in prep*) The Shape of Sound made step by step (*Supplementary file*). *Methods in Ecology and Evolution*.

Thanks for reading!

Quick note for beginners in R

This supplementary file was generated using `rmarkdown` package (Allaire et al. 2017) from R platform (R Core Team, 2017). R is a free and open software without any guarantees. It works under an open code and has its own language. It is not the scope of

this supplementary file to explain the caveats of such language as it has been done elsewhere (e.g. Crawley, 2012).

The full code employed in our analysis is displayed herein, which can be intimidating for those unfamiliar with R's environment. Therefore, whenever a code is about to be presented, a quick explanation of it will precede. In addition, graphical outputs are preceded by the code used to produce it, thus enabling anyone to reproduce it using their own data.

Packages required

```
R.Version()$version.string; R.Version()$platform

## [1] "R version 3.4.1 (2017-06-30)"

## [1] "x86_64-w64-mingw32"

library(seewave)
packageVersion("seewave")

## [1] '2.1.0'

library(tuneR)
packageVersion("tuneR")

## [1] '1.3.2'

library(plot3D)
packageVersion("plot3D")

## [1] '1.1.1'

library(abind)
packageVersion("abind")

## [1] '1.4.5'

library(knitr)
packageVersion("knitr")

## [1] '1.17'

library(captioner)
```

```
library(reshape2)
packageVersion("reshape2")

## [1] '1.4.3'

library(geomorph)
packageVersion("geomorph")

## [1] '3.0.5'

library(shapes)
packageVersion("shapes")

## [1] '1.2.0'
```

A brief introduction

Geometric Morphometrics applied to bioacoustical analysis is a new approach aimed for the direct comparison between calls of different species. To support our study on empirical data, we analyzed the calls of 30 species of *Physalaemus* Fitzinger (Anura, Leptodactylidae), a genus currently composed of 48 recognized species (Frost, 2017) divided in two major clades: the *P. cuvieri* and the *P. signifer* clades (Lourenço et al., 2016). Even though this work is focused on the advertisement calls of a single genus of frogs, the methods are described as an open code that can be adapted and applied on any given sound, depending on the limitations and objectives one might be interested in. For more information, please see Rocha & Romano (*in prep*).

Our work encompassed two distinct methods, both derived from the Geometric Morphometrics discipline. *SoundShape* analysis consisted in the acquisition of semilandmarks using energy quantiles within a curve of relative amplitude. Semilandmark coordinates are then used for standard Geometric Morphometric analysis, in which a Principal Component Analysis (PCA) are conducted for reduction in data dimensionality. Additionally, we also replicate the *eigensound* approach (MacLeod et al., 2013), which calculates semilandmark coordinates from the sound surface of a spectrogram output. The three dimensional coordinates are then submitted to PCA as well. In other words, *SoundShape* computes the two dimensional (2D) shape of a sonogram, whereas the *eigensound* deals with the three dimensional (3D) surface of it. Here we compare *eigensound* and *SoundShape* approaches in order to present a

comprehensive study on this matter and make easier to anyone interested in applying these methods to implement it.

As with any bioacoustical analysis, the first and foremost step in sound shape analysis is the careful selection of sound units from which analysis will be conducted. Although there is no universal concept of a homologous unit of biological sound encompassing the majority of calling organisms, each higher taxon has its own approaches for homologous sound comparison. The criteria can be mechanical or physiological, such as the syllables generated during a complete opening and closure of the forewings in orthopterans (Ragge & Reynolds, 1998; Robillard & Desutter-Grandcolas, 2011), the notes produced in a complete airflow cycle between the lungs and the vocal sac in frogs (McLister et al., 1995; Robillard et al., 2006), or the mechanical sounds produced by Neotropical manakin birds during lek displays (Prum, 1998). In other cases, the criteria is mostly behavioral, such as the long and complex bird songs typically employed on the attraction of mates or deflection of competitors (Päckert et al., 2003; Cate, 2004; Marler, 2004), or the bat echolocation calls, which are used for navigation and identification of objects in the absence of light (Schnitzler & Kalko, 2001; MacLeod et al., 2013).

Regardless of the criteria, it is pivotal that the selection of sounds reflect the homology between units (Robillard et al., 2006; Cap et al., 2008; Robillard & Desutter-Grandcolas, 2011), thus assuring a meaningful comparison of acoustic signals. In addition, one must bear in mind that overlapping frequencies in the spectrogram can have a great impact on the point coordinates acquired by either SoundShape or eigensound analysis. Therefore, the recordings employed in any sound shape study must account for optimal signal to noise ratio and no overlap with background individuals (see sections below for shape definition).

SoundShape 2D analysis: Semilandmarks from a curve of relative amplitude

Detection of relative amplitude contours

Each of the 30 *Physalaemus* species had one to four individuals sampled according to call availability from bioacoustical collections. Each individual had five calls selected and stored as separate wave files. In order to respect the homology between calls of different

species, herein we adopt a physiological definition of “note”, in which a note is defined as the total amount of sound energy produced by a single airflow cycle through the larynx, between the lungs and the vocal sacs (McLister et al., 1995). This mechanism of sound production is observed in most species of frogs, involving the same muscles, cartilages and membranes. Therefore, the sound produced by this airflow cycle can be considered the homologous unit of frogs call, and will be referred hereafter simply as note (Robillard et al., 2006). Note definition was not always easy though, given that this approach require an observer to watch the frog during calling activity. Thus, whenever the physiological concept became too arbitrary, we followed the note definition from Köhler et al. (2017) (see Rocha & Romano *in prep* for details).

To perform both *SoundShape* and *eigensound* analysis in a comparable way, spectrograms must be as much stereotyped as possible. Given that the slightest graphical change may incur in biased results, all spectrogram parameters must be clearly specified. Time and frequency (*i.e.* X and Y-axis, respectively) for instance, were set to range from 0 to 4 s, and from 0 to 6 kHz, respectively. This choice encompassed the longer notes (*i.e.* *Physalaemus olfersii*) and the ones with higher values of frequency (*i.e.* *P. bokermanni*), assuring that all images would be on the same scale. Additionally, the window length was set with FFT=512, 70% overlap and 44100 Hz sampling rate, which are standard parameters recommended for bioacoustical analysis (Köhler et al., 2017). We will provide more details on how these settings impact the spectrogram output once we move into the *eigensound* approach.

In order to illustrate how the acquisition of semilandmarks was conducted, we selected a multipulsed note from *Physalaemus centralis* (Steindachner, 1864) (Figure 1) and one harmonically structured from *P. kroyeri* (Reinhardt and Lütken, 1862) (Figure 2), both with medium duration (approximately 0.6 s).

Note: These sounds were edited for better visualization, with total duration reduced to 1 s, frequency ranges restricted from 0 to 4 kHz and notes displayed at spectrograms' center.

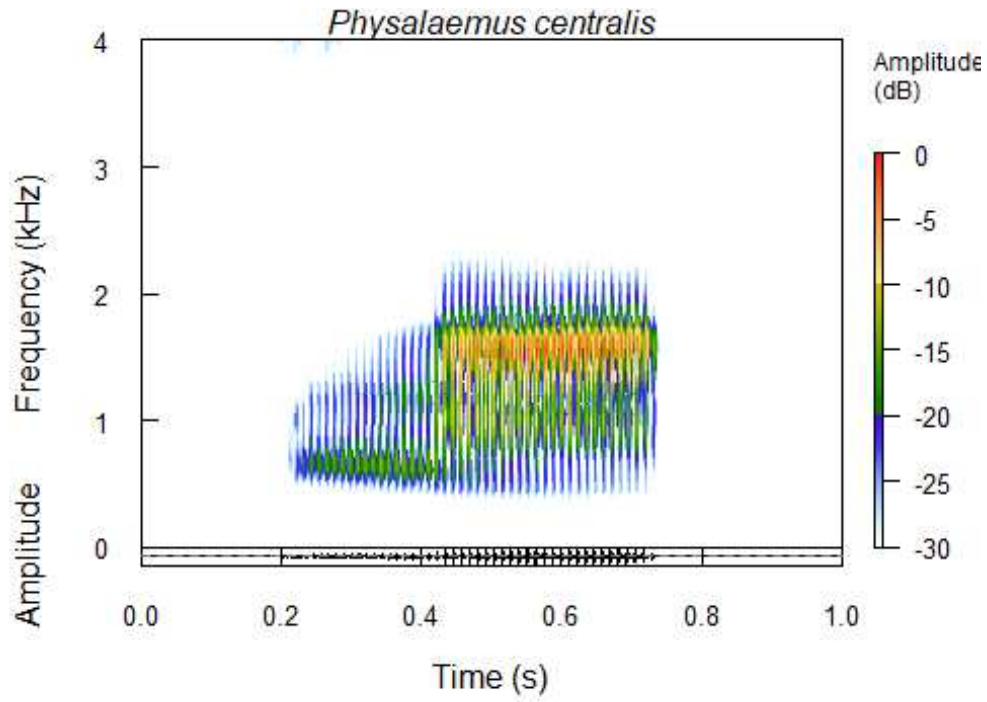


Figure 1: Spectrogram (above) and oscillogram (below) views of the advertisement call of *Physalaemus centralis*.

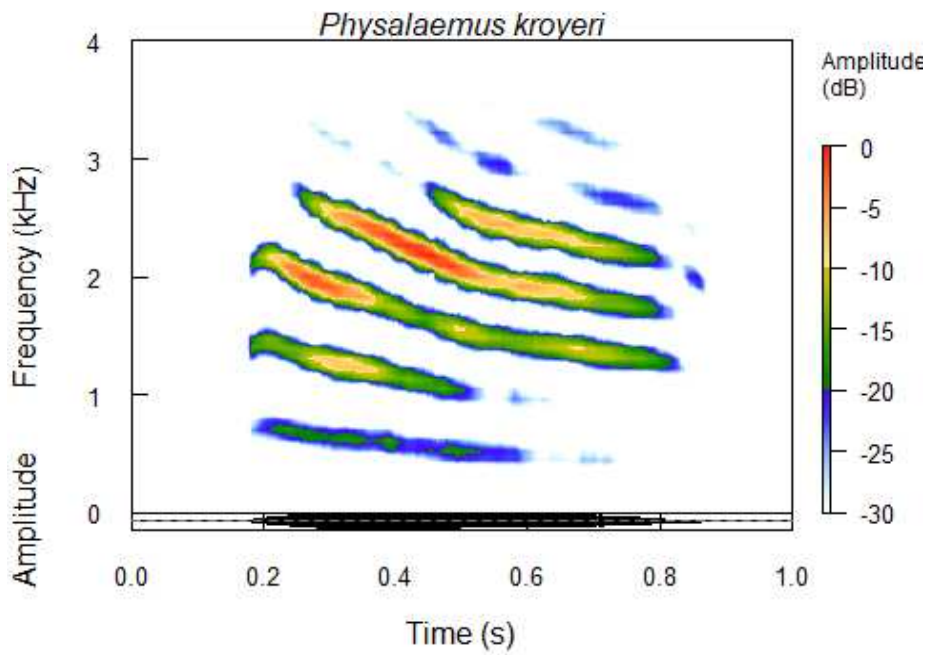


Figure 2: Spectrogram (above) and oscillogram (below) views of the advertisement call of *Physalaemus kroyeri*.

Both *SoundShape* and *eigensound* are methods aimed for recordings made at uncontrolled conditions. However, caller's enthusiasm and its distance from the recording device can affect spectrogram's output (Gerhardt & Huber, 2002, Charif et al. 2010, Köhler et al., 2017). On the other hand, these factors do not influence the nature of the signal nor its wave form, it only affects the amount of information gathered by the

recorder. Therefore, the homologous comparison between shapes required spectrograms more stereotyped than standardized plots.

If the spectrogram is a two dimensional plot of a three dimensional form (third dimension – Z axis – represented by amplitude as color gradient), one could think of it as a satellite picture of a mountain, with surrounding valleys covered with fog (*i.e.* background noise). Therefore, the picture only display the altitudes above the fog, even though the fog itself do not influence the mountains' landscape (*i.e.* sound shape) (Figure 3 and 4).

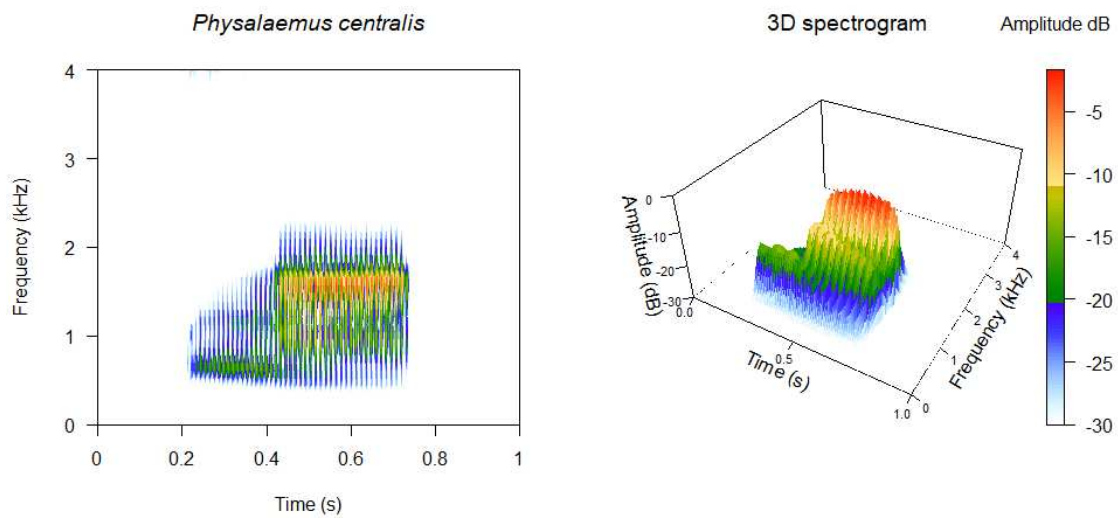


Figure 3: Standard two dimensional spectrogram of the advertisement call of *Physalaemus centralis* (left) and the three dimensional equivalent of it (right).

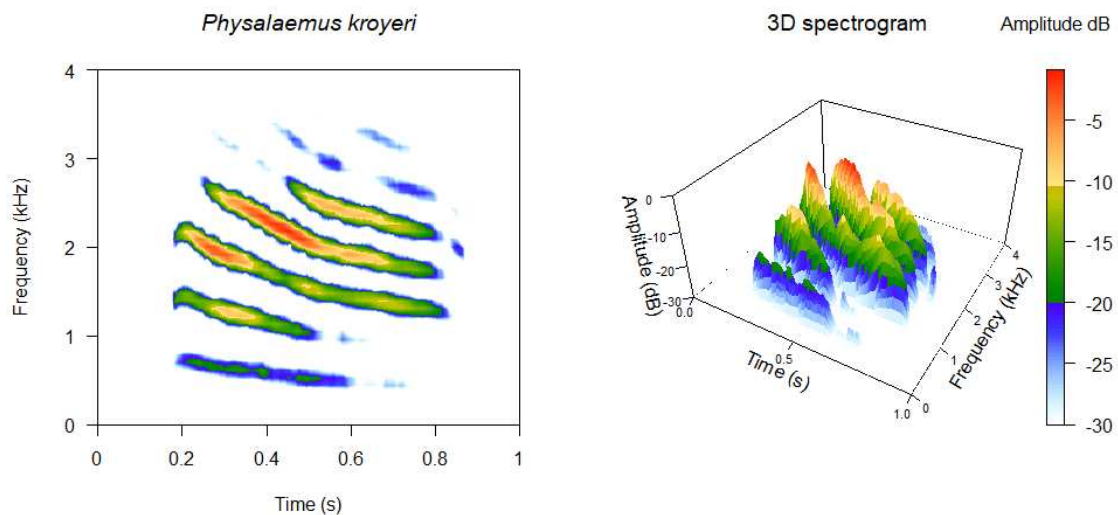


Figure 4: Standard two dimensional spectrogram of the advertisement call of *Physalaemus kroyeri* (left) and the three dimensional equivalent of it (right).

Thus, if one wishes to compare the landscapes of different mountains surrounded by various levels of fog, the comparison would have to be made above an curve of relative altitude in which all mountains are clearly pictured without loss of information. Provided that each mountain has a different altitude (*i.e.* amplitude level), the simpler way is to consider the altitude backwards, with the highest peak set as 0, and the predefined altitude curve set as an negative value.

In our study, the curve of relative amplitude was set at -25 dB, which was experimentally defined as the amplitude value that encompass the general shape of a call with minimum influence of background noise. MacLeod et al. (2013) employed a similar approach for the *eigensond* analysis; in their words, “is the sonic equivalent of placing a specimen on a black background prior to collecting a digital image of its form”. By doing this, we managed to compare the most energetic part of each note without loss of the general shape of the call (Figure 5 and 6).

R code: The code below include functions `readWave` and `spectro` from `seewave` package. The first is used to input `wave` files into R environment by assigning the result of `readWave` to an R object (*i.e.* `P.eph` and `P.kro`). The second computes a spectrogram view of the `wave` object created, which can also be assigned to a different R object. The `spectro` object include a time `$time` and frequency `$freq` coordinates, plus a matrix of amplitude values `$amp` that can produce a three dimensional surface plot using `persp3D` function (`plot3D` package). However the matrix `$amp` also include `-Inf` values, which causes an error with `persp3D`. Therefore we create a loop to replace any values below -30 dB or `-Inf` before plotting the three dimensional surface. In addition, we call the function `par` to enable both two and three dimensional spectrograms to be plotted side by side. Ultimately, the relative amplitude contours are plotted differently in each spectrogram plot. On the 2D spectrogram, it is added by simply calling the arguments `cont=TRUE` and `contlevels=seq(-dBlevel, -dBlevel, 1)`, in which `dBlevel` refers to the absolute value of relative amplitude. Whereas in the 3D spectrogram the curve is added separately using `contour3D` function. For details on the arguments of any function, type on R’s console the name of the function preceded by ‘?’ (*e.g.* `?spectro`) and execute it.

```
# Read wave file
P.eph <- readWave("FNJV-0031235_P-centralis_BB1_003.wav")
P.kro <- readWave("FNJV-0032047_P-kroyeri_I11_002.wav")
```

```

# Amplitude level to use as relative amplitude contour.
Defined experimentaly
dBlevel = 25

# Physalaemus centralis
par(mfrow=c(1,2)) # Plot side by side
s.eph <- spectro(P.eph, osc=FALSE, f=44100, wl=512,
ovlp=70, tlim=c(0,1), flim=c(0,4), cont=TRUE, contlevels=
seq(-dBlevel, -dBlevel, 1), lwd=0.8, grid=FALSE, scale=FALSE,
main=expression(italic('Physalaemus centralis'))

for(i in 1:length(s.eph$amp)){if(s.eph$amp[i] == -Inf |
s.eph$amp[i] <= -30)
{s.eph$amp[i] <- -30}} # Set background at -30 dB

persp3D(x=s.eph$time, y=s.eph$freq, z=t(s.eph$amp), theta=30,
phi=35, resfac = 2, r=3, expand=0.5, scale=TRUE, axes=TRUE,
ticktype="detailed", nticks=4, col=spectro.colors(n=100),
cex.axis=0.7, xlab="Time (s)", ylab="Frequency (kHz)",
zlab="Amplitude (dB)", clab=expression('Amplitude dB'), main=
expression('3D contours'))

contour3D(x=s.eph$time, y=s.eph$freq, colvar=t(s.eph$amp),
z=-25, plot=T, add=T, addbox=F, col="black", lwd=1.2, nlevels=2,
dDepth=0.2)

```

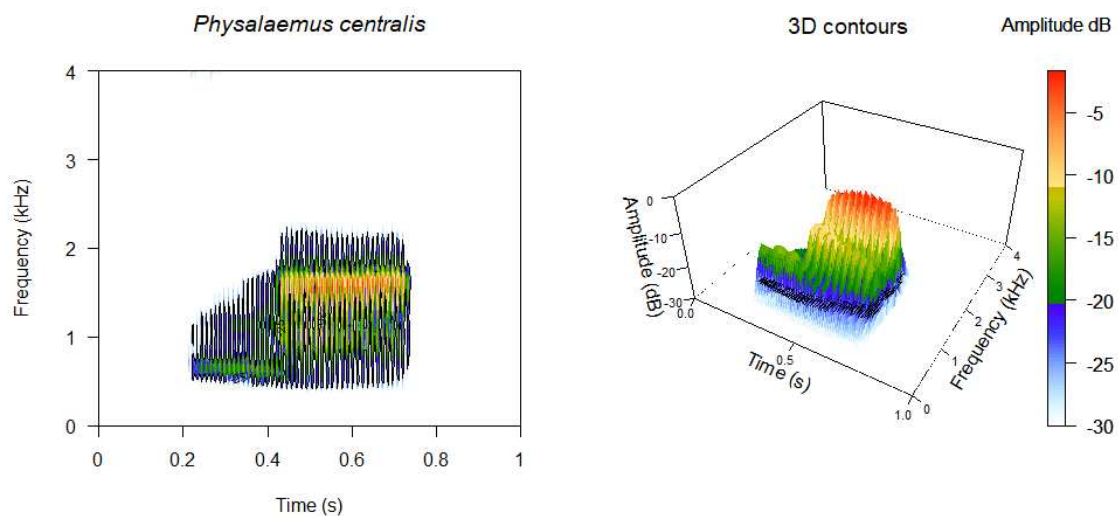


Figure 5: Two (left) and three dimensional spectrogram views (right) of the advertisement call of *Physalaemus centralis* with relative amplitude contours at -25 dB.

```

# Physalaemus kroyeri
par(mfrow=c(1,2))

s.kro <- spectro(P.kro, osc=FALSE, f=44100, wl=512, ovlp=70,
tlim=c(0,1), flim=c(0,4), cont=TRUE, contlevels=seq(-dBlevel,
-dBlevel, 1), lwd=1, grid=FALSE, scale=FALSE,
main=expression(italic('Physalaemus kroyeri'))

for(i in 1:length(s.kro$amp)){if(s.kro$amp[i] == -Inf |
s.kro$amp[i] <= -30)
{s.kro$amp[i] <- -30}} # Set background at -30 dB

persp3D(x=s.kro$time, y=s.kro$freq, z=t(s.kro$amp), theta=30,
phi=35, resfac = 3, r=3, expand=0.5, scale=TRUE, axes=TRUE, tick
type="detailed", nticks=4, col=spectro.colors(n=100),
cex.axis=0.7, xlab="Time (s)", ylab="Frequency (kHz)",
zlab="Amplitude (dB)", clab=expression('Amplitude dB'),
main=expression('3D contours'))

contour3D(x=s.kro$time, y=s.kro$freq, colvar=t(s.kro$amp),
z=-25, plot=T, add=T, addbox = F, col="black", lwd=1.4,
nlevels=2, dDepth=0.2)

```

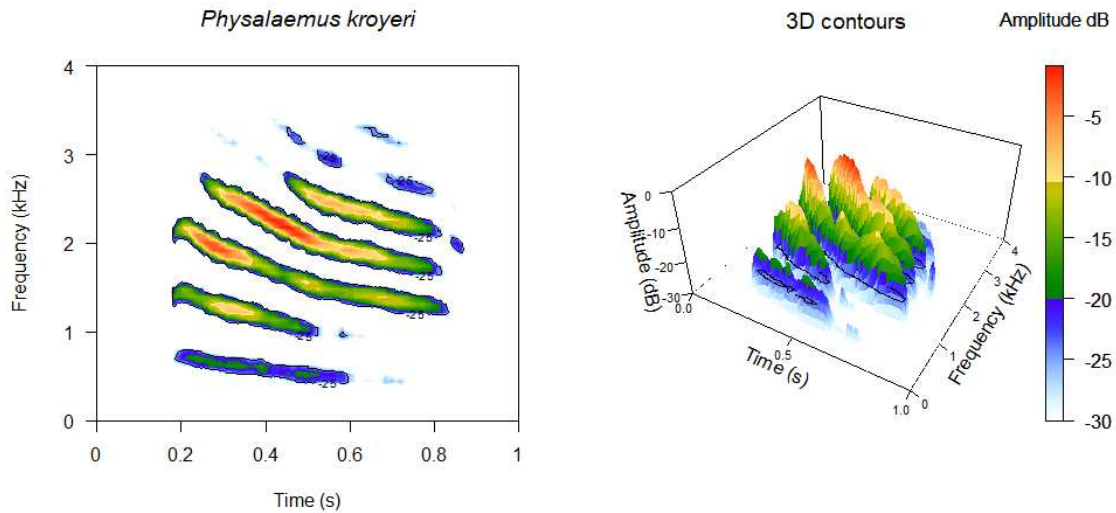


Figure 6: Two (left) and three dimensional spectrogram views (right) of the advertisement call of *Physalaemus kroyeri* with relative amplitude contours at -25 dB.

Next, we select the longest contour, which will represent the shape of sound for each note. The selected contour will also be used to unveil initial and final time values within the curve, as well as minimum and maximum frequency values (and time values associated to it).

Note: Contour selection in multipulsed notes (e.g. *Physalaemus centralis*) is quite straightforward, as these usually have a single curve delimiting the whole note. On the other hand, harmonically structured notes often require careful inspection *a posteriori* of the curve selected, given that the longest contour may vary between harmonics within the same individual.

R code: Contour selection uses the `spectro` object created on the previous code chunk. First, all contours for a given relative amplitude `dBlevel` are assigned to an R object using `contourLines` function. The length of each contour detected is then stored in a numeric vector and used to select the longest contour. This procedure is repeated for each `spectro` object.

```
# Selection of longest contour

# Physalaemus centralis
con.eph <- contourLines(x=s.eph$time, y=s.eph$freq,
z=t(s.eph$amp), levels=seq(-dBlevel,-dBlevel,1))
n.eph <- numeric(length(con.eph))
for(i in 1:length(con.eph)) n.eph[i] <- length(con.eph[[i]]$x)
n.eph.max <- which.max(n.eph)
con.eph <- con.eph[[n.eph.max]]

# Physalaemus kroyeri
con.kro <- contourLines(x=s.kro$time, y=s.kro$freq,
z=t(s.kro$amp), levels=seq(-dBlevel,-dBlevel,1))
n.kro <- numeric(length(con.kro))
for(i in 1:length(con.kro)) n.kro[i] <- length(con.kro[[i]]$x)
n.kro.max <- which.max(n.kro)
con.kro <- con.kro[[n.kro.max]]
```

R code: Once these values are acquired, we can now work within the square limits of the selected contour (Figure 7 and 8). To do so, we select the initial and final time values within the curve. Similarly, we also select the minimum and maximum frequency values (and the correspondent time values to be used later).

```
# Initial and final time values within relative amplitude
contour

# Physalaemus centralis
```

```

t.eph.beg <- min(con.eph$x)
t.eph.end <- max(con.eph$x)

# Physalaemus kroyeri
t.kro.beg <- min(con.kro$x)
t.kro.end <- max(con.kro$x)

# Minimum and maximum frequency values within relative
amplitude contour

# Physalaemus centralis
f.eph.min <- min(con.eph$y) # Minimum frequency
t.eph.min <- con.eph$x[con.eph$y==f.eph.min] # Time value
associated

f.eph.max <- max(con.eph$y) # Maximum frequency
t.eph.max <- con.eph$x[con.eph$y==f.eph.max] # Time value
associated

# Physalaemus kroyeri
f.kro.min <- min(con.kro$y) # Minimum frequency
t.kro.min <- con.kro$x[con.kro$y==f.kro.min] # Time value
associated

f.kro.max <- max(con.kro$y) # Maximum frequency
t.kro.max <- con.kro$x[con.kro$y==f.kro.max] # Time value
associated

```

R code: The square limits of the longest contour can be visualized along with the spectrogram plot and the longest contour selected. First, we call the `spectro` plot without the relative amplitude curves. The longest contour is plotted separately using `polygon` function. Initial and final time values, minimum and maximum frequency values are used as the square limits of a second `polygon`.

```

spectro(P.eph, osc=FALSE, grid=FALSE, scale=FALSE, f=44100,
wl=512, ovlp=70, tlim=c(0,1), flim=c(0,4),
main=expression(italic('Physalaemus centralis'))
polygon(x=con.eph$x, y=con.eph$y, lwd=1) # Selected contour

```

```

polygon(x=c(t.eph.beg, t.eph.beg, t.eph.end, t.eph.end),
y=c(f.eph.min, f.eph.max, f.eph.max, f.eph.min), lty="dashed",
lwd=1.3) # Square limits

```

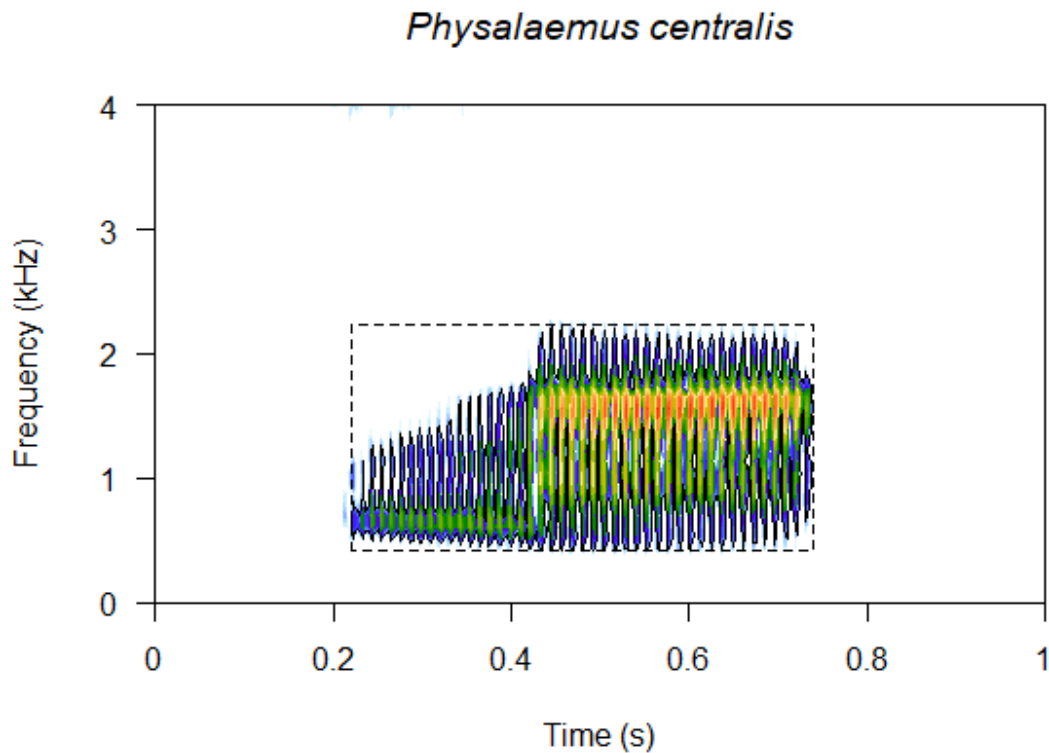


Figure 7: Spectrogram view of the advertisement call of *Physalaemus centralis* with longest curve of relative amplitude at -25 dB and square limits drawn around contour.

```

spectro(P.kro, osc=FALSE, grid=FALSE, scale=FALSE, f=44100,
wl=512, ovlp=70, tlim=c(0,1), flim=c(0,4),
main=expression(italic('Physalaemus kroyeri')))
polygon(x=con.kro$x, y=con.kro$y, lwd=2.5) # Selected contour
polygon(x=c(t.kro.beg, t.kro.beg, t.kro.end, t.kro.end),
        y=c(f.kro.min, f.kro.max, f.kro.max, f.kro.min),
lty="dashed", lwd=1.3) # Square limits

```

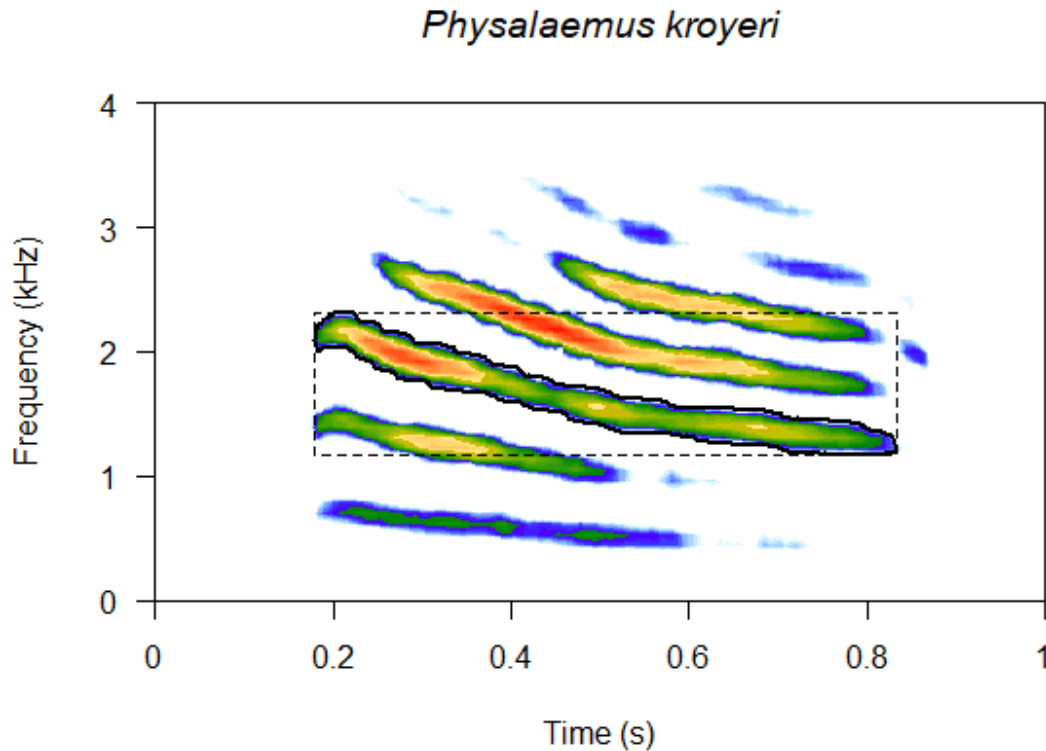


Figure 8: Spectrogram view of the advertisement call of *Physalaemus kroyeri* with longest curve of relative amplitude at -25 dB and square limits drawn around contour.

Acquisition of semilandmarks coordinates to use in Geometric Morphometric analysis

Now that we have a contour and square limits to work with (Figure 7 and 8), next step is the acquisition of landmarks to use in the Geometric Morphometric (GM) analysis. This step aim to transform the selected contour into a comparable matrix of point coordinates, which also reduces the amount of variables and optimize the performance in the later steps of GM analysis. This is done by simplifying the curve of relative amplitude into a predefined set of points, which must be the same number of points for all sounds analyzed.

Landmarks (LM) are often referred as “homologous points” and defined as discrete anatomical loci that can be recognized as the same point in all specimens in the study (Zelditch et al. 2012). However, when dealing with spectrograms, it is not feasible to obtain discrete homologous points among notes. The relative amplitude contours, on the other hand, are complex curves that encompass comparable information about the shape of sound and, as aforementioned, a single contour can be objectively treated as

homologous. Therefore, if we manage to acquire enough points to describe such contour, then those points could be considered analogue or equal to semilandmarks (SM), thus being comparable between any taxonomical level (Zelditch et al. 2012). Even though the SMs are not homologous per se, the curve described by them is.

In frogs, the spectral properties of a call (*i.e.* frequency measures such as the dominant frequency) are typically less variable than temporal ones (*e.g.* call duration) (Gerhardt 1991, Gerhardt & Huber 2002, Köhler et al. 2017). They are often classified as static and frequently associated with species recognition. Thus, phylogenetically related species are expected to present more similar call frequencies, which would result in spectrograms with similar shapes. Since spectrograms are a graphical representation of sound energy (*i.e.* amplitude) in the time x frequency domain, the shape of sound will change as the energy shifts within the call. That shift is mainly caused by morphological changes in the calling apparatus of frogs and, to some extent, reflects the phylogenetic distance among them.

Traditionally, spectral properties included in bioacoustical analysis account for the peak of energy within the call (*i.e.* dominant frequency) and the frequency bandwidth that encompass the majority of sound energy (*i.e.* 90% frequency bandwidth) (Köhler et al. 2017). These spectral measures are calculated from a Power Spectrum, which in turn represents the relative amplitude for each frequency component and can also be used to acquire frequency values for each energy quantiles. In our approach, we considered homologous the energy quantiles within the square limits of notes, implying that *SoundShape* will compute how the shifts in sound energy alter the shape of sound.

Note: Environmental conditions (*e.g.* air temperature, humidity, presence of other males/females) might also impact the shape of sound, especially because changes in call duration directly impacts the spectrogram output. Therefore, it is recommended that comparative analysis are conducted with recordings made under similar environmental conditions to prevent biased results.

In our analysis, we managed to acquire 16 semilandmarks within the longest contour of any given call, which is the maximum number of points that did not cause an computational error in R (*e.g.* narrow banded notes resulted in overlapped SMs). Each SM represent an energy quantile and its coordinates are given by the correspondent frequency value and the cross-correlated time value within the curve of relative amplitude.

The minimum and maximum quantiles (SM-min, SM-max), for instance, are represented by the minimum and maximum frequency values within the longest contour. These values have been calculated previously (*e.g.* `t.eph.min` and `f.eph.min`) and can be thought as *x* and *y* coordinates for the quantiles of 0 (SM-min) and 100% energy (SM-max) (Figure 9 and 10).

R code: Since the SM-min and SM-max have already been calculated, we can add them as points on a spectrogram plot.

```
spectro(P.eph, osc=F, grid=F, scale=F, f=44100, wl=512,
ovlp=70, tlim=c(0,1), flim=c(0,4),
main=expression(italic('Physalaemus kroyeri')))
polygon(x=con.eph$x, y=con.eph$y, lwd=1) # Selected contour
points(x=c(t.eph.min,t.eph.max), y=c(f.eph.min,f.eph.max),
pch=21, col="black", bg="red", cex=1.1)
text(x=c(t.eph.min, t.eph.max), y=c(f.eph.min, f.eph.max),
labels=c("SM-min", "SM-max"), cex=0.75, pos=c(1,3))
```

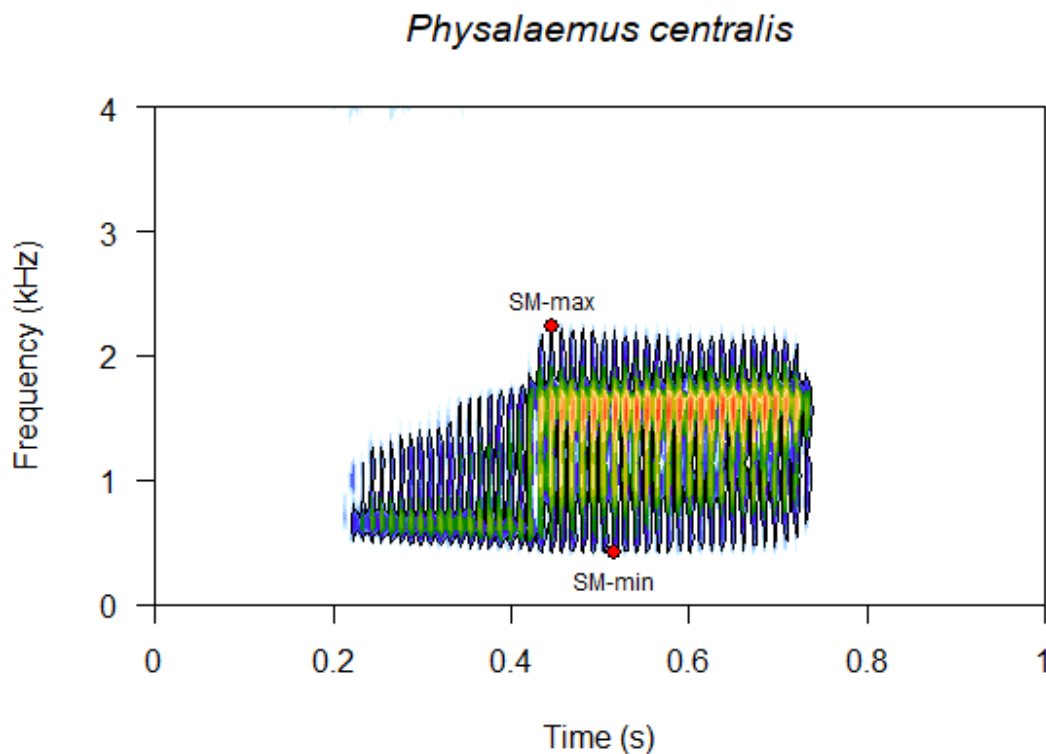


Figure 9: Spectrogram view of the advertisement call of *Physalaemus centralis* with longest curve of relative amplitude at -25 dB and two semilandmarks (red dots) plotted using minimum (SM-min) and maximum (SM-max) frequency values within the curve.

```
spectro(P.kro, osc=F, grid=F, scale=F, f=44100, wl=512,
ovlp=70, tlim=c(0,1), flim=c(0,4),
```

```

main=expression(italic('Physalaemus kroyeri'))
  polygon(x=con.kro$x, y=con.kro$y, lwd=2) # Selected
  contour
  points(x=c(t.kro.min,t.kro.max),y=c(f.kro.min,f.kro.max),
pch=21, col="black", bg="red", cex=1.1)
  text(x=c(t.kro.min, t.kro.max),y=c(f.kro.min, f.kro.max),
      labels=c("SM-min", "SM-max"), cex=0.75, pos=c(1,3))

```

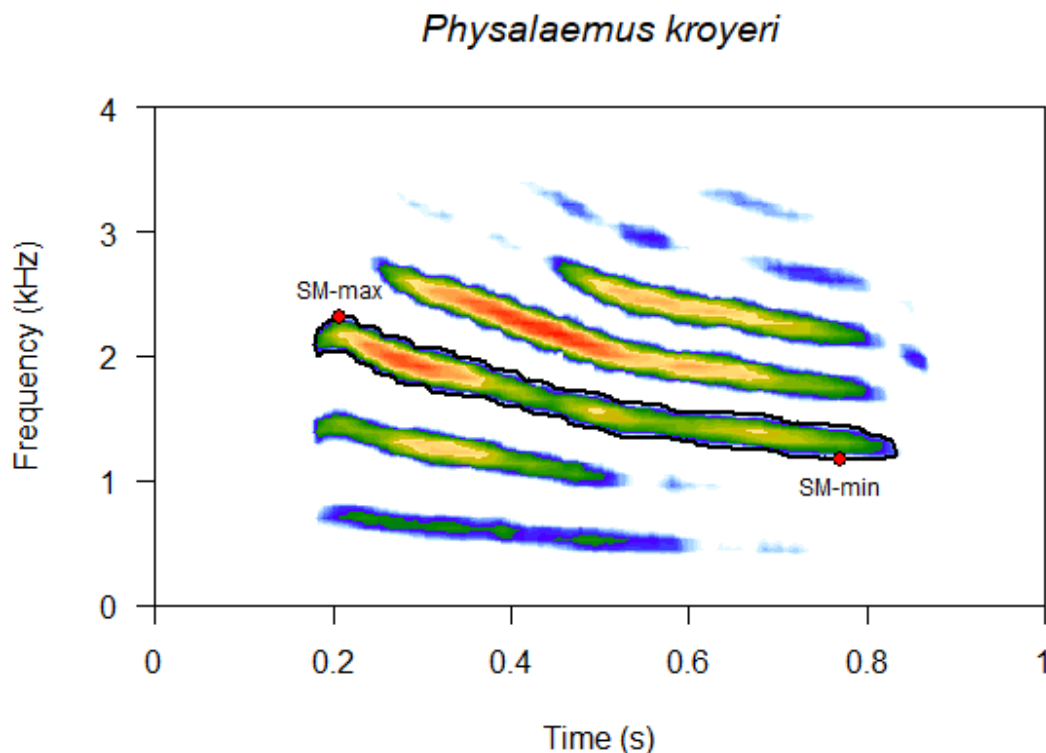


Figure 10: Spectrogram view of the advertisement call of *Physalaemus kroyeri* with longest curve of relative amplitude at -25 dB and two semilandmarks (red dots) plotted using minimum (SM-min) and maximum (SM-max) frequency values within the curve.

The remaining 14 SMs are the cross-correlation between energy quantiles (given as frequency values) and their associated time values.

R code: The following steps will require new functions `zC` and `frequencies`. `zC` computes the zero-crossing for a curve of relative amplitude and has been written by Jérôme Sueur. The latter is an adaptation of `specprop` (`seewave` package) and returns a `data.frame` of frequency values according to the energy quantiles `EQ` desired. In order to use these functions, one must either store them as an R script (e.g. "`zC-function.R`") and then `source` it into R's environment, or paste and run it directly in R's console. Arguments `spec`, `f`, `mel` and `flim` are the same as in `specprop()`. For details, please see `?specprop()`.

```
# Function to compute zero-crossing for a curve of relative  
amplitude (by Jérôme Sueur)
```

```
ZC <- function(x) {  
  sign <- ifelse(x >= 0, 1, -1)  
  res <- abs(diff(sign))  
  return(res)  
}
```

Table 1: Arguments to be used by frequencies function.

Arguments	Description
spec	a data set resulting of a spectral analysis obtained with spec or meanspec (not in dB).
f	sampling frequency of spec (in Hz).
mel	a logical, if TRUE the (htk-)mel scale is used.
flim	a vector of length 2 to specify the frequency limits of the analysis (in kHz).
EQ	a vector of energy quantiles to be calculated (with $0 < EQ < 1$)

```
# Function to acquire frequency values for each energy  
quantile EQ
```

```
frequencies <- function (spec, f = NULL, mel = FALSE,  
flim = NULL, EQ= c(0.05, 0.15, 0.3, 0.5, 0.7, 0.85, 0.95))  
{  
  fhz <- f  
  if (!is.null(f) & mel) {f <- 2 * mel(f/2)}  
  if (is.null(f)) {  
    if (is.vector(spec))  
      stop("'f' is missing")  
    else if (is.matrix(spec))  
      f <- spec[nrow(spec),1]*2000*nrow(spec)/(nrow(spec) - 1)  
  }  
  
  if (is.matrix(spec)) {  
    freq <- spec[, 1]  
    freq = freq * 1000  
    spec <- spec[, 2]  
  }  
  
  L <- length(spec)  
  wl <- L * 2  
  if (any(spec < 0))  
    stop("The frequency spectrum to be analysed should not be  
in dB")
```

```

if (!is.null(flim)) {
  if (flim[1] < 0 || flim[2] > fhz/2)
    stop("'flim' should range between 0 and f/2")
  if (mel)
    flim <- mel(flim * 1000)/1000
}
else {flim <- c(0, f/2000)}

spec <- spec[(flim[1] * 1000 * wl/f):(flim[2]* 1000 * wl/f)]
amp <- spec/sum(spec)
cumamp <- cumsum(amp)
freq <- seq(from = flim[1] * 1000, to = flim[2] * 1000,
length.out = length(spec))

for(i in EQ) # Frequency values according to desired energy
quantiles
{ if(!exists("Qlist")) {Qlist <- data.frame()}
  if(exists("Qlist")) {
    Qtemp<-data.frame("Quantiles" = paste("EQ-", i, sep=""),
                      "Frequency values" = freq[
length(cumamp[cumamp <= i]) + 1])
    Qlist <- rbind(Qlist, Qtemp)
    rm(Qtemp)
  }
}
results <- Qlist
}

```

Function `frequencies` calculates energy quantiles using the result of `spec` or `meanspec`. These functions, in turn, calculate the relative amplitude for each frequency component (*i.e.* Power Spectrum) in a given sound wave. Since our analysis is focused on the longest contour within a call, both `spec` and `frequencies` must be calculated within the limits of the contour (delimited by a square, see Figure 7 and 8), reducing the influence of sound energies outside the curve of relative amplitude (*i.e.* the homologous curve). By doing this, the calculation of `spec` may still include energy outside the curve; however the noise in the data is reduced. Additionally, sampling frequency f and the window length wl must be the same defined *a priori*.

R code: In order to acquire frequency values for each energy quantile EQ , one must first source the `frequencies` function. Then, a power spectrum is calculated using `spec` and stored into an R object. The `spec` object will be used by `frequencies` to acquire the frequency values for each energy quantile EQ desired.

Note: `spec` only enables to set the time limits for the power spectrum calculation, whereas `frequencies` only enable the frequency limits, thus encompassing the square limits of the selected curve of relative amplitude.

```
# Source frequencies function into R's environment
source("insert.filepath.here/frequencies.R")

# Acquiring frequency values for each energy quantile

# Physalaemus centralis
pwr.spec.eph <- spec(P.eph, f=44100, wl=512, from=t.eph.beg,
to=t.eph.end, plot=F)
fvalues.eph <- frequencies(pwr.spec.eph, f=44100,
flim=c(f.eph.min, f.eph.max),
EQ= c(0.05, 0.15, 0.30, 0.50, 0.70, 0.85, 0.95)) # EQ = energy
quantiles
fvalues.eph$Frequency.values <- (fvalues.eph$Frequency.values)
/1000 #frequency values are given in Hz by function frequencies;
we divide it by 1000 to deal with them in kHz
print(fvalues.eph)

##   Quantiles Frequency.values
## 1   EQ-0.05           0.6263332
## 2   EQ-0.15           0.8737299
## 3   EQ-0.3            1.1230594
## 4   EQ-0.5            1.4226414
## 5   EQ-0.7            1.5521382
## 6   EQ-0.85           1.6507103
## 7   EQ-0.95           1.9135693

# Physalaemus kroyeri
pwr.spec.kro <- spec(P.kro, f=44100, wl=512, from=t.kro.beg,
to=t.kro.end, plot=F)
fvalues.kro <- frequencies(pwr.spec.kro, f=44100,
flim=c(f.kro.min, f.kro.max),
```

```
EQ= c(0.05, 0.15, 0.30, 0.50, 0.70, 0.85, 0.95)) # EQ = energy
quantiles
  fvalues.kro$Frequency.values <- (fvalues.kro$Frequency.values)
/1000 # frequency values
  # are given in Hz by function frequencies; we divide it by
1000 to deal with them in kHz
  print(fvalues.kro)

##   Quantiles Frequency.values
## 1   EQ-0.05           1.258346
## 2   EQ-0.15           1.471019
## 3   EQ-0.3            1.751013
## 4   EQ-0.5            1.966746
## 5   EQ-0.7            2.107508
## 6   EQ-0.85           2.200839
## 7   EQ-0.95           2.272750
```

Once acquired the frequency values for each energy quantile, we must find the time value associated with each frequency quantile. These values are the intersection between the calculated frequency and the curve of relative amplitude. In other words, each frequency quantile is a Y coordinate that can be represented by a line. That line will always have two intersections with the curve of relative amplitude. Each point of intersection is a X coordinate to go along with the calculated Y . In the example below, the frequency value associated with the 50% energy quantile is used to acquire two time values that intersect with the curve of relative amplitude (Figure 11 and 12).

Note: This procedure produce some dependence among SMs defined by the same frequency value, and consequently reduce the variance contained in each pair of SMs. However, this is a straightforward criteria to easily capture part of the curve of relative amplitude.

R code: Similarly as the last code chunk, we first source `ZC` function into R's environment. Since we aim to use the frequency value associated with the 50% energy quantile, we store this value as a separate R object. The `ZC` function is then used to identify and store the points of intersection between our selected frequency value and the curve of relative amplitude. These will be used to find the first and last point of intersection (in relation to time X -axis), which can be plotted on the spectrogram along with the SM-min and SM-max.

```

# Source ZC function into R's environment
source("insert.filepath.here/ZC.R")

# Physalaemus centralis

# Selecting the frequency value associated with the 50% energy
quantile
f50.eph <- (fvalues.eph$Frequency.values[fvalues.eph$Quantiles
=="EQ-0.5"])

# Find the positions where the selected frequency intersects
with the curve of relative amplitude
zc.res.eph <- ZC(con.eph$y - f50.eph)
t.pos.eph <- which(zc.res.eph==2)
t.eph <-c(min(con.eph$x[t.pos.eph]),max(con.eph$x[t.pos.eph]))
# Select first and last points of intersection

# Plotting the results of the cross-correlation
spectro(P.eph, osc=F, grid=F, scale=F, f=44100, wl=512,
ovlp=70, tlim=c(0,1), flim=c(0,4),
main=expression('Semilandmarks at 50% energy quantile (' *italic
('Physalaemus centralis')*')'))
polygon(x=con.eph$x, y=con.eph$y, lwd=2) # Selected contour
text(x=0.05, y=f50.eph, cex=0.6, labels="EQ = 50%")
clip(0.105, 1, 0, 4)
abline(h=f50.eph, lty="dashed")
points(x=t.eph, y=c(f50.eph, f50.eph), pch=21, col="black",
bg="red", cex=1.1) # for each frequency value, we have two time
values associated
points(x=c(t.eph.min,t.eph.max), y=c(f.eph.min,f.eph.max),
pch=21, col="black", bg="red", cex=1.1) # SM-min and SM-max
text(x=c(t.eph.min, t.eph.max), y=c(f.eph.min, f.eph.max),
labels=c("SM-min", "SM-max"), cex=0.6, pos=c(1,3))

```

Semilandmarks at 50% energy quantile (*Physalaemus centralis*)

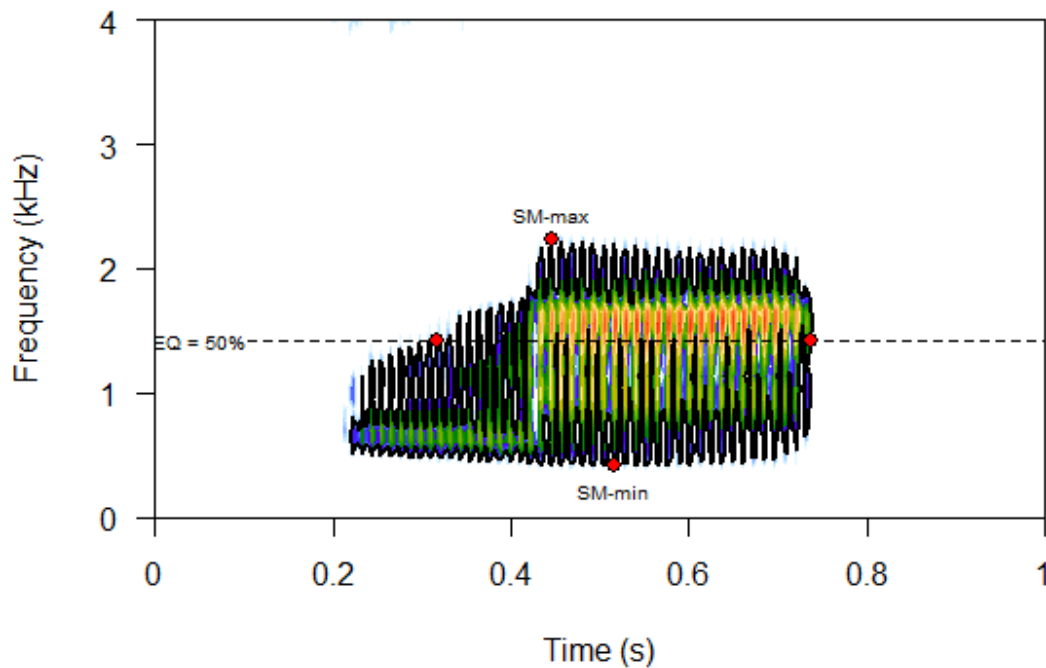


Figure 11: Spectrogram view of the advertisement call of *Physalaemus centralis* with longest curve of relative amplitude at -25 dB, minimum (SM-min) and maximum (SM-max) frequency values, and semilandmarks plotted as cross correlation between contour and frequency value representing 50% energy quantile. Semilandmarks represented by red dots.

```
# Physalaemus kroyeri

# Selecting the frequency value associated with the 50% energy
quantile
f50.kro <- (fvalues.kro$Frequency.values[fvalues.kro$Quantiles
=="EQ-0.5"])

# Find the positions where the selected frequency intersects
with the curve of relative amplitude
zc.res.kro <- ZC(con.kro$y - f50.kro)
t.pos.kro <- which(zc.res.kro==2)
t.kro <- c(min(con.kro$x[t.pos.kro]), max(con.kro$x[t.pos.kro]
)) # Select first and last points of intersection

# Plotting the results of the cross-correlation
spectro(P.kro, osc=F, grid=F, scale=F, f=44100, wl=512,
ovlp=70, tlim=c(0,1), flim=c(0,4),main=expression('Semilandmarks
at 50% energy quantile (' *italic('Physalaemus kroyeri')*')))
```

```

polygon(x=con.kro$x, y=con.kro$y, lwd=2) # Selected contour
text(x=0.05, y=f50.kro, cex=0.6, labels="EQ = 50%")
clip(0.105, 1, 0, 4)
abline(h=f50.kro, lty="dashed")
points(x=t.kro, y=c(f50.kro, f50.kro), pch=21, col="black",
bg="red", cex=1.1) # for each frequency value, we have two time
values associated

points(x=c(t.kro.min,t.kro.max), y=c(f.kro.min,f.kro.max),
pch=21, col="black", bg="red", cex=1.1) # SM-min and SM-max
text(x=c(t.kro.min, t.kro.max), y=c(f.kro.min, f.kro.max),
labels=c("SM-min", "SM-max"), cex=0.6, pos=c(1,3))

```

Semilandmarks at 50% energy quantile (*Physalaemus kroyeri*)

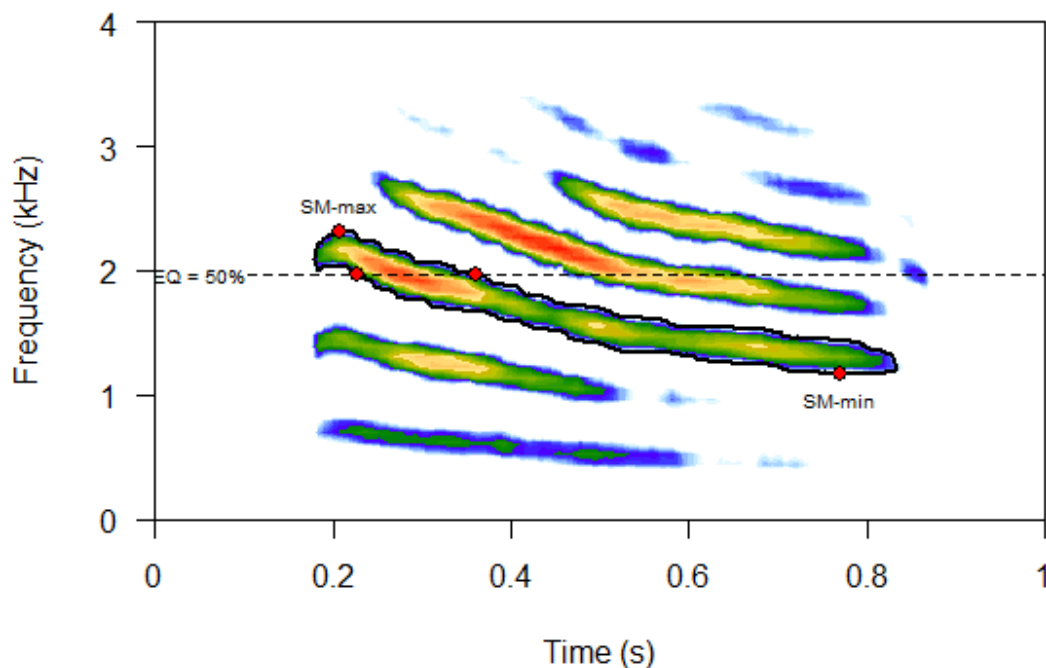


Figure 12: Spectrogram view of the advertisement call of *Physalaemus kroyeri* with longest curve of relative amplitude at -25 dB, minimum (SM-min) and maximum (SM-max) frequency values, and semilandmarks plotted as cross correlation between contour and frequency value representing 50% energy quantile. Semilandmarks represented by red dots.

R code: We can now create a loop to acquire time values for each energy quantile. The `for` loop will compute the points of intersection between the selected contour and each frequency value in the `data.frame` created by `frequencies`. Time values are stored as vectors at the end of the loop.

```

# Loop to acquire time values for each frequency quantile

```

```

# Physalaemus centralis
for(i in seq_along(fvalues.eph$Frequency.values)) {
  zc.res.eph<-ZC(con.eph$y-(fvalues.eph$Frequency.values[i]))
  t.pos.eph <- which(zc.res.eph==2)

  if(!exists("t1.eph")|!exists("t2.eph")){ # Creating vectors
to store the time values
    t1.eph <- vector(mode="integer", length=length(fvalues.eph
$Frequency.values))
    t2.eph <- vector(mode="integer", length=length(fvalues.eph
$Frequency.values))
  }

  if(exists("t1.eph")|exists("t2.eph")){
    t1.eph[i] <- min(con.eph$x[t.pos.eph])
    t2.eph[i] <- max(con.eph$x[t.pos.eph])
  }
}

# Physalaemus kroyeri
for(i in seq_along(fvalues.kro$Frequency.values)) {
  zc.res.kro<- ZC(con.kro$y-(fvalues.kro$Frequency.values[i]))
  t.pos.kro <- which(zc.res.kro==2)

  if(!exists("t1.kro")|!exists("t2.kro")){ # Creating vectors
to store the time values
    t1.kro <- vector(mode="integer", length=length(fvalues.kro
$Frequency.values))
    t2.kro <- vector(mode="integer", length=length(fvalues.kro
$Frequency.values))
  }

  if(exists("t1.kro")|exists("t2.kro")){
    t1.kro[i] <- min(con.kro$x[t.pos.kro])
    t2.kro[i] <- max(con.kro$x[t.pos.kro])
  }
}

```

R code: Now we bind the vectors created by the loop and store them on a `data.frame` along with the frequency values `fvalues`. These will be plotted along with the selected curve and the SM-min and SM-max (Figure 13 and 14).

```
# Coordinates for plotting

# Physalaemus centralis
SM1.eph <- data.frame(cbind(fvalues.eph, "Time"= t1.eph))
SM2.eph <- data.frame(cbind(fvalues.eph, "Time"= t2.eph))

# Physalaemus kroyeri
SM1.kro <- data.frame(cbind(fvalues.kro, "Time"= t1.kro))
SM2.kro <- data.frame(cbind(fvalues.kro, "Time"= t2.kro))

# Plotting the 16 semilandmarks according to each energy
quantile

# Physalaemus centralis
spectro(P.eph, osc=F, grid=F, scale=F, f=44100, wl=512, ovlp=70,
tlim=c(0,1), flim=c(0,4),
main=expression(italic('Physalaemus centralis')))
polygon(x=con.eph$x, y=con.eph$y, lwd=2) # Selected contour
text(x=0.95, y=fvalues.eph$Frequency.values, cex=0.6,
labels=c("EQ = 5%", "", "EQ = 30%", "", "EQ = 70%", "", "EQ = 95%"))
text(x=0.05, y=fvalues.eph$Frequency.values, cex=0.6,
labels=c("", "EQ = 15%", "", "EQ = 50%", "", "EQ = 80%", ""))
clip(0.105, 1, 0, 4)
abline(h=fvalues.eph$Frequency.values[c(2,4,6)], lty="dashed")
clip(0, 0.895, 0, 4)
abline(h=fvalues.eph$Frequency.values[c(1,3,5,7)],
lty="dashed")
points(x=c(SM1.eph$Time, SM2.eph$Time),
y=c(SM1.eph$Frequency.values, SM2.eph$Frequency.values), pch=21,
col="black", bg="red", cex=1.1) # Plotting semilandmarks
points(x=c(t.eph.min, t.eph.max), y=c(f.eph.min, f.eph.max),
pch=21, col="black", bg="red", cex=1.1) # SM-min and SM-max
text(x=c(t.eph.min, t.eph.max), y=c(f.eph.min, f.eph.max),
labels=c("SM-min", "SM-max"), cex=0.6, pos=c(1,3))
```

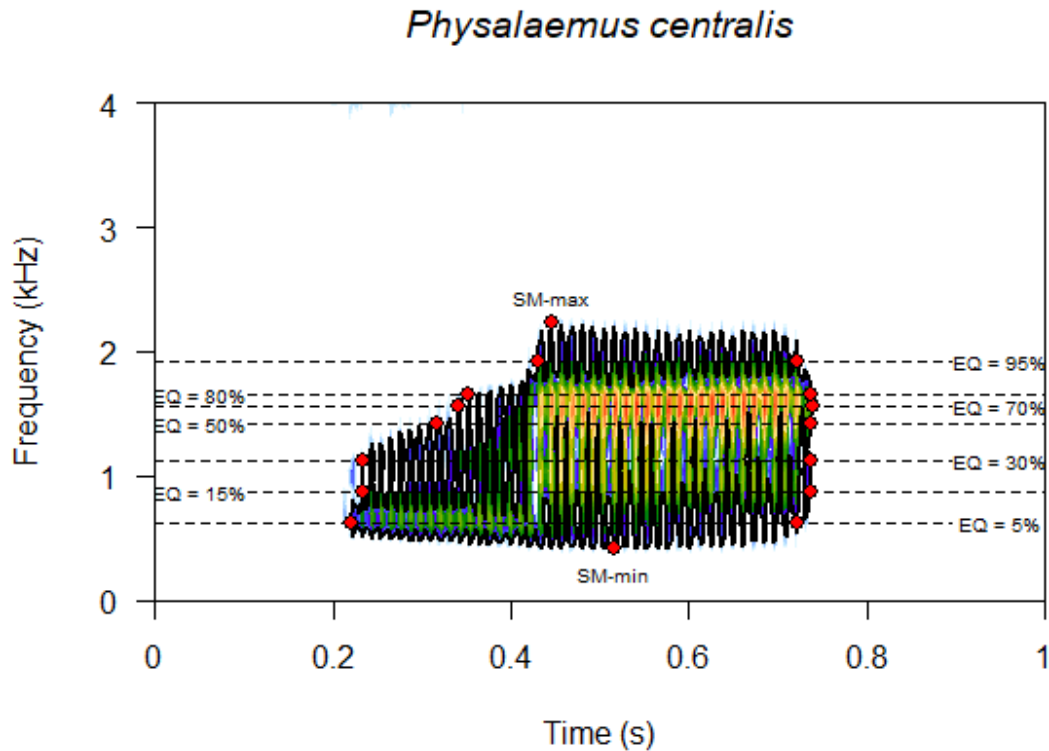


Figure 13: Spectrogram view of the advertisement call of *Physalaemus centralis* with longest curve of relative amplitude at -25 dB, SM-min and SM-max, and 14 semilandmarks resulting from the cross correlation between contour and seven frequency values representing energy quantiles. Semilandmarks represented by red dots.

```
# Physalaemus kroyeri
spectro(P.kro,osc=F,grid=F, scale=F, f=44100, wl=512, ovlp=70,
tlim=c(0,1), flim=c(0,4),
main=expression(italic('Physalaemus kroyeri'))
polygon(x=con.kro$x, y=con.kro$y, lwd=2) # Selected contour
text(x=0.95, y=fvalues.kro$Frequency.values, cex=0.6,
labels=c("EQ = 5%", "", "EQ = 30%", "", "EQ = 70%", "", "EQ = 95%"))
text(x=0.05, y=fvalues.kro$Frequency.values, cex=0.6,
labels=c("", "EQ = 15%", "", "EQ = 50%", "", "EQ = 80%", ""))
clip(0.105, 1, 0, 4)
abline(h=fvalues.kro$Frequency.values[c(2,4,6)], lty="dashed")
clip(0, 0.895, 0, 4)
abline(h=fvalues.kro$Frequency.values[c(1,3,5,7)],
lty="dashed")
points(x=c(SM1.kro$Time, SM2.kro$Time),
y=c(SM1.kro$Frequency.values, SM2.kro$Frequency.values),pch=21,
col="black", bg="red", cex=1.1) # Plotting semilandmarks
points(x=c(t.kro.min,t.kro.max), y=c(f.kro.min,f.kro.max),
```

```
pch=21, col="black", bg="red", cex=1.1) # SM-min and SM-max
text(x=c(t.kro.min, t.kro.max), y=c(f.kro.min, f.kro.max),
labels=c("SM-min", "SM-max"), cex=0.6, pos=c(1,3))
```

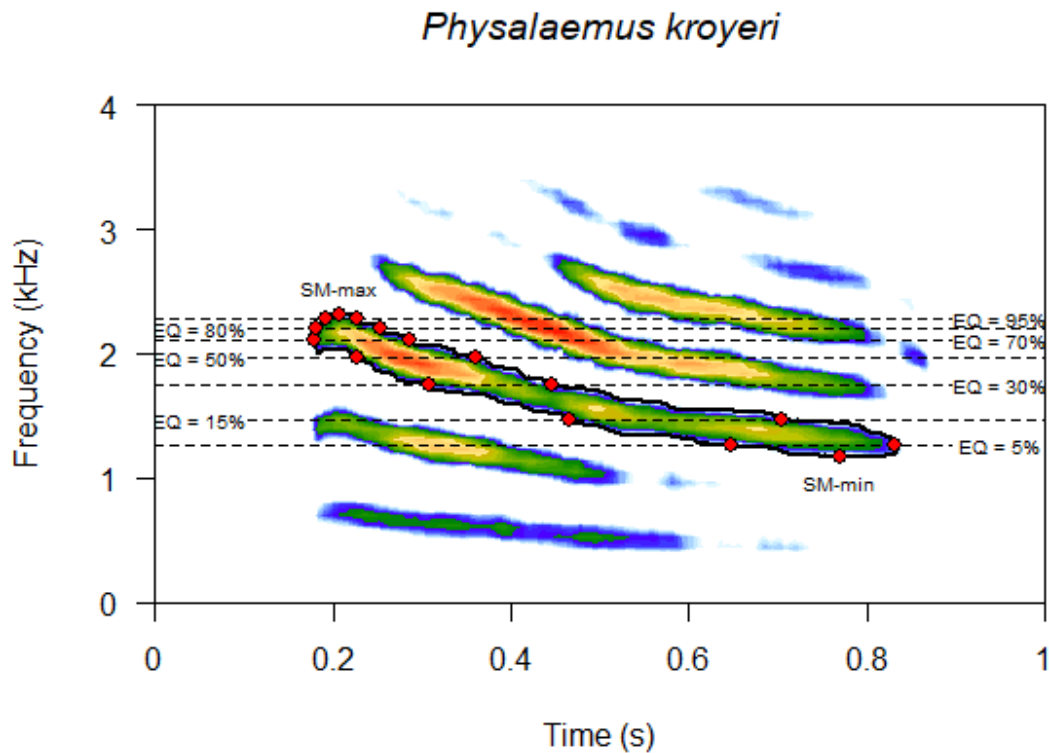


Figure 14: Spectrogram view of the advertisement call of *Physalaemus kroyeri* with longest curve of relative amplitude at -25 dB, SM-min and SM-max, and 14 semilandmarks resulting from the cross correlation between contour and seven frequency values representing energy quantiles. Semilandmarks represented by red dots.

Lastly, we include the SM-min and SM-max in the `data.frame` to compose the complete set of 16 semilandmarks (Figure 15 and 16). SMs are ordered and numbered anticlockwise, starting from the SM-max.

R code: Time and frequency coordinates for SM-min and SM-max are stored into a `data.frame`. A new vector with semilandmark numbers is created. The complete set of semilandmarks is binded into a single `data.frame` and plotted on the spectrogram along with the selected contour.

Note: The set of semilandmarks resulting from the cross correlation with initial time values (*i.e.* SM1) is re-ordered so that semilandmark numbering and plotting will be anticlockwise.

```
# Store X and Y coordinates SM-min and SM-max in a data.frame
```

```

# Physalaemus centralis
SM.min.eph <- data.frame("Quantiles" = "EQ-min",
"Frequency.values"= f.eph.min, "Time"= t.eph.min)
SM.max.eph <- data.frame("Quantiles" = "EQ-max",
"Frequency.values"= f.eph.max, "Time"= t.eph.max)

# Physalaemus kroyeri
SM.min.kro <- data.frame("Quantiles" = "EQ-min",
"Frequency.values"= f.kro.min, "Time"= t.kro.min)
SM.max.kro <- data.frame("Quantiles" = "EQ-max",
"Frequency.values"= f.kro.max, "Time"= t.kro.max)

# Re-order SM1 so that points plotting will be anticlockwise
SM1.eph <- SM1.eph[order(SM1.eph[,2], decreasing=T),]
SM1.kro <- SM1.kro[order(SM1.kro[,2], decreasing=T),]

# Semilandmark numbers
SMs.numbers <- as.factor(1:16)

# Complete data.frame with coordinates of x and y
SM.eph<-cbind(SMs.numbers,data.frame(rbind(SM.max.eph,SM1.eph,
SM.min.eph, SM2.eph))) # P. centralis
print(SM.eph)

##      SMs.numbers Quantiles Frequency.values      Time
## 1             1      EQ-max      2.2305464 0.4452297
## 7             2      EQ-0.95      1.9135693 0.4296780
## 6             3      EQ-0.85      1.6507103 0.3523377
## 5             4      EQ-0.7       1.5521382 0.3415564
## 4             5      EQ-0.5       1.4226414 0.3176878
## 3             6      EQ-0.3       1.1230594 0.2328429
## 2             7      EQ-0.15      0.8737299 0.2326291
## 11            8      EQ-0.05      0.6263332 0.2210926
## 12            9      EQ-min      0.4117938 0.5159011
## 13           10      EQ-0.05      0.6263332 0.7225015
## 21           11      EQ-0.15      0.8737299 0.7362347
## 31           12      EQ-0.3       1.1230594 0.7362966
## 41           13      EQ-0.5       1.4226414 0.7372901

```

```
## 51          14      EQ-0.7          1.5521382 0.7384748
## 61          15      EQ-0.85         1.6507103 0.7378094
## 71          16      EQ-0.95         1.9135693 0.7220915

SM.kro<-cbind(SMs.numbers,data.frame(rbind(SM.max.kro,SM1.kro,
SM.min.kro, SM2.kro))) # P. kroyeri

print(SM.kro)

##      SMs.numbers Quantiles Frequency.values      Time
## 1              1      EQ-max          2.314061 0.2084806
## 7              2      EQ-0.95         2.272750 0.1914672
## 6              3      EQ-0.85         2.200839 0.1805907
## 5              4      EQ-0.7          2.107508 0.1793125
## 4              5      EQ-0.5          1.966746 0.2261484
## 3              6      EQ-0.3          1.751013 0.3074205
## 2              7      EQ-0.15         1.471019 0.4664311
## 11             8      EQ-0.05         1.258346 0.6466431
## 12             9      EQ-min          1.171135 0.7703180
## 13            10      EQ-0.05         1.258346 0.8303887
## 21            11      EQ-0.15         1.471019 0.7046062
## 31            12      EQ-0.3          1.751013 0.4452297
## 41            13      EQ-0.5          1.966746 0.3604240
## 51            14      EQ-0.7          2.107508 0.2862191
## 61            15      EQ-0.85         2.200839 0.2544170
## 71            16      EQ-0.95         2.272750 0.2261484

# Physalaemus centralis
spectro(P.eph, osc=F,grid=F,scale=F, f=44100, wl=512, ovlp=70,
tlim=c(0,1), flim=c(0,4),
main=expression(italic('Physalaemus centralis'))
polygon(x=con.eph$x, y=con.eph$y, lwd=2) # Selected contour
points(x=SM.eph$Time, y=SM.eph$Frequency.values, pch=21,
col="black", bg="red", cex=1.1)
text(x=SM.eph$Time, y=SM.eph$Frequency.values, cex=0.6,
labels=SM.eph$SMs.numbers, pos=c(3,2,3, 3,3,2, 2,2,1,
rep(4, dim(SM2.eph)[1]))))
```

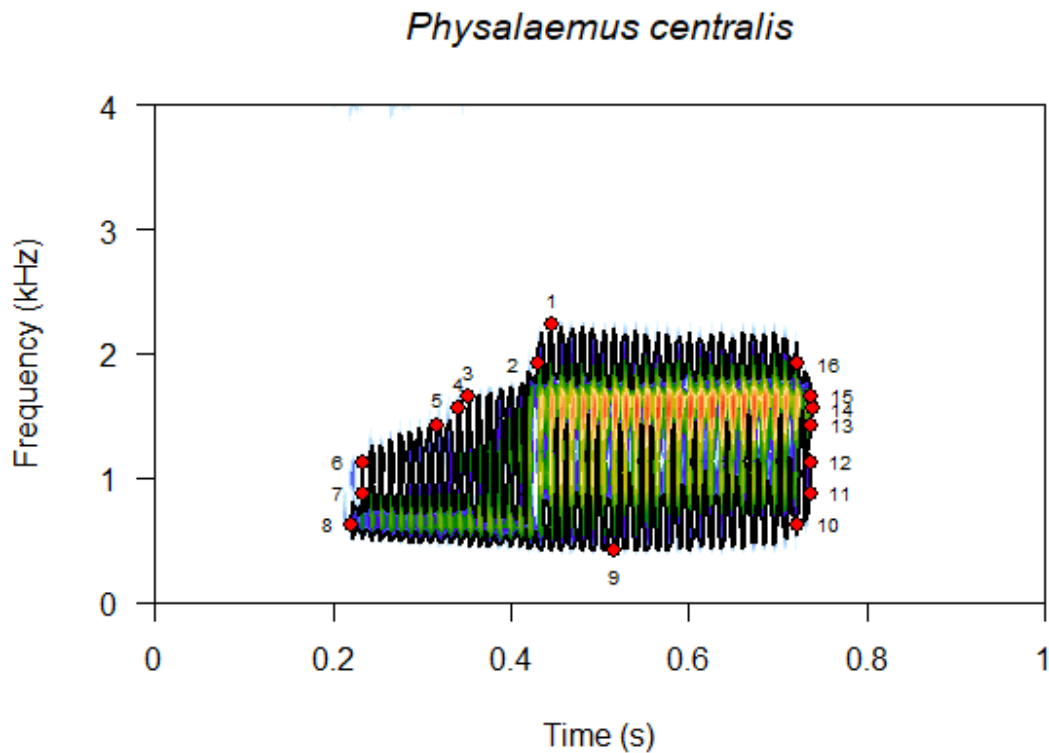


Figure 15: Spectrogram view of the advertisement call of *Physalaemus centralis* with complete set of 16 semilandmarks plotted as cross correlation between longest curve of relative amplitude at -25 dB, and frequency values representing energy quantiles within the contour. Semilandmarks represented by red dots.

```
# Physalaemus kroyeri
spectro(P.kro, osc=F, grid=F, scale=F, f=44100, wl=512, ovlp=70,
tlim=c(0,1), flim=c(0,4),
main=expression(italic('Physalaemus kroyeri'))
polygon(x=con.kro$x, y=con.kro$y, lwd=2) # Selected contour
points(x=SM.kro$Time, y=SM.kro$Frequency.values, pch=21,
col="black", bg="red", cex=1.1)
text(x=SM.kro$Time, y=SM.kro$Frequency.values, cex=0.6,
labels=SM.kro$SMs.numbers,
pos=c(3,3,2, 2,1,1, 1,1,1, 4,3,3, 3,3,3, 3))
```

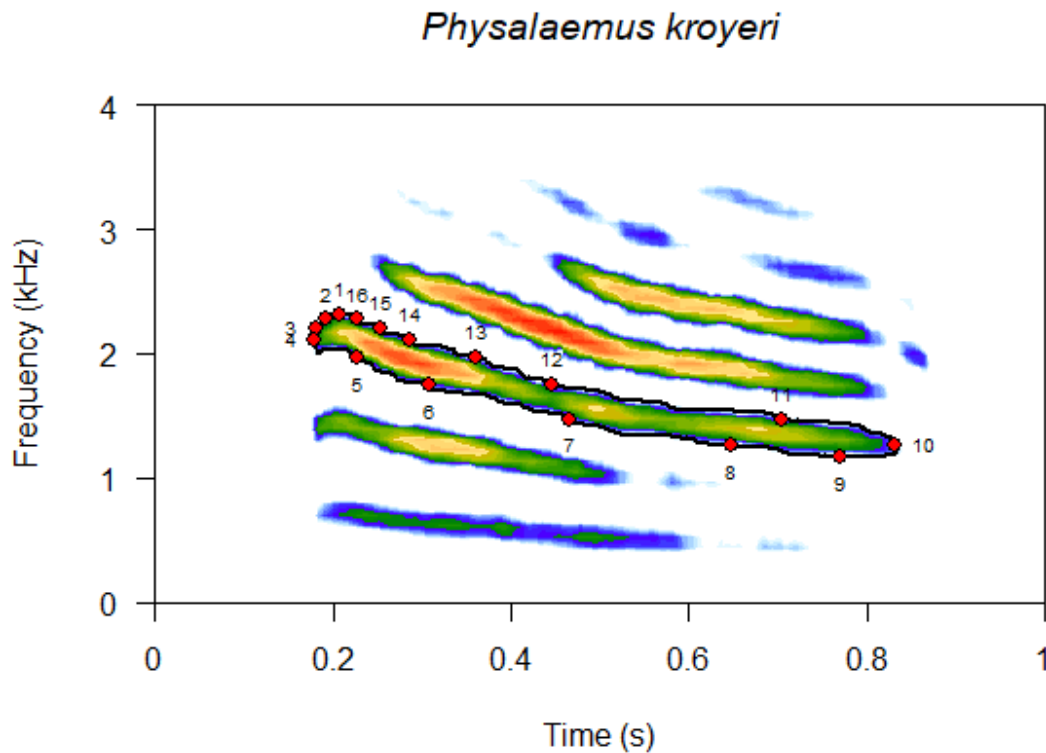


Figure 16: Spectrogram view of the advertisement call of *Physalaemus kroyeri* with complete set of 16 semilandmarks plotted as cross correlation between longest curve of relative amplitude at -25 dB, and frequency values representing energy quantiles within the contour. Semilandmarks represented by red dots.

Now that we have a `data.frame` with semilandmark coordinates, we can even discard the spectrogram itself and henceforward proceed with the analysis regarding only the shape of sound.

R code: Semilandmark points and wireframe polygon are plotted using the coordinates stored on the `data.frame` in the last code chunk. The plot itself is called as an empty page by calling the argument `type='n'`.

```
# Physalaemus centralis
plot(x=SM.eph$Time, y=SM.eph$Frequency.values, type='n',
xlim=c(0,1), ylim=c(0,4), xlab="Time coordinates",
ylab="Frequency coordinates",
main=expression(italic('Physalaemus centralis'))
polygon(x=SM.eph$Time, y=SM.eph$Frequency.values)
points(x=SM.eph$Time, y=SM.eph$Frequency.values, pch=21,
col="black", bg="red", cex=1.2)
text(x=SM.eph$Time, y=SM.eph$Frequency.values, cex=0.6,
labels=SM.eph$SMs.numbers, pos=c(3,2,3, 3,3,2, 2,2,1,
rep(4, dim(SM2.eph)[1])))
```

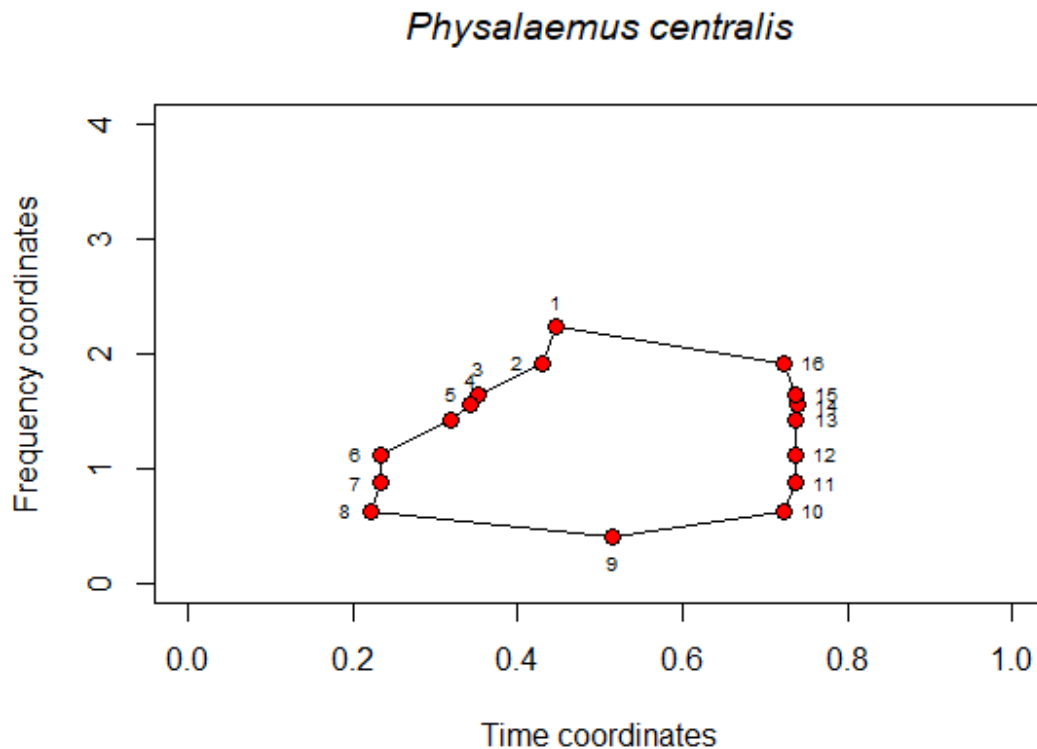


Figure 17: Complete set of 16 semilandmarks (red dots) and wireframe representing the sound shape of the advertisement call of *Physalaemus centralis*. Semilandmarks were acquired through cross correlation between longest curve of relative amplitude and frequency values representing energy quantiles within the contour.

```
# Physalaemus kroyeri
plot(x=SM.kro$Time, y=SM.kro$Frequency.values, xlim=c(0,1),
ylim=c(0,4),xlab="Time coordinates",
ylab="Frequency coordinates",
main=expression(italic('Physalaemus kroyeri')))
polygon(x=SM.kro$Time, y=SM.kro$Frequency.values)
points(x=SM.kro$Time, y=SM.kro$Frequency.values, pch=21,
col="black", bg="red", cex=1.2)
text(x=SM.kro$Time, y=SM.kro$Frequency.values, cex=0.6,
labels=SM.kro$SMs.numbers,
pos=c(3,3,2, 2,1,1, 1,1,1, 4,3,3, 3,3,3, 3))
```

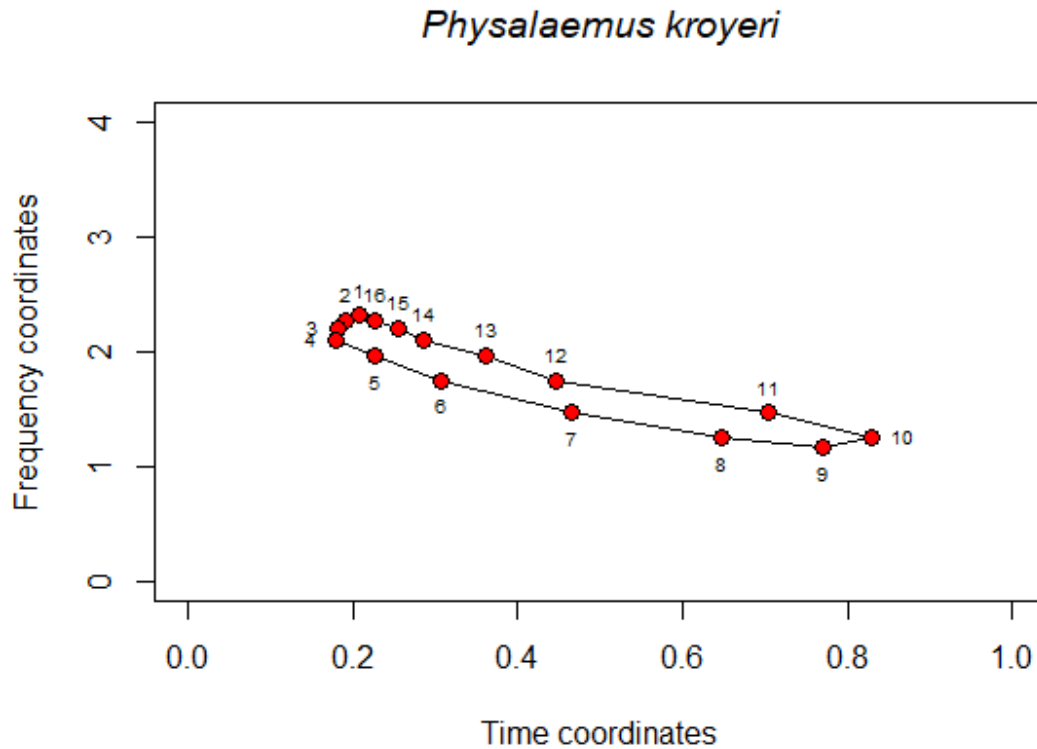


Figure 18: Complete set of 16 semilandmarks (red dots) and wireframe representing the sound shape of the advertisement call of *Physalaemus kroyeri*. Semilandmarks were acquired through cross correlation between longest curve of relative amplitude and frequency values representing energy quantiles within the contour.

SoundShape function: Acquiring 2D semilandmarks from multiple wave files

In order to facilitate the comparative analysis of multiple notes, we introduce the new function `SoundShape`, whose default will compute 16 semilandmarks as result of the cross-correlation between seven energy quantiles `EQ` and a curve of relative amplitude `dBlevel`.

The results may be assigned to an R object and used in the Geometric Morphometric steps of the analysis. Alternatively, it is also possible to export semilandmark coordinates as a `tps` file to be used by numerous softwares of geometric analysis of shape, such as the TPS series (Rohlf, 2015). Additionally, one might also export spectrogram images with contour and semilandmarks plotted, which is useful for semilandmark inspection and/or build figures for publication. Images can be exported as `jpeg` (compressed image) and `tiff` (uncompressed image) file formats .

R code: In order to use `SoundShape` one must first import it into the R environment. As described for the `frequencies` and `ZC` functions, this can be done by pasting the whole code below into R's console and then executing it, or by sourcing the function into R (*i.e.* `source("insert.filepath.here/SoundShape.R")`).

Table 2: Arguments to be used by `SoundShape` function.

Arguments	Description
<code>f</code>	Sampling frequency of wave files (in Hz). By default = 44100.
<code>wl</code>	Length of the window for the analysis. By default = 512.
<code>ovlp</code>	Overlap between two successive windows (in %) for increased spectrogram resolution. By default = 70.
<code>dBlevel</code>	Absolute amplitude value to be used as relative amplitude contour, which will serve as reference for semilandmark acquisition.
<code>flim</code>	Modifications of the frequency Y-axis limits. Vector with two values in kHz.
<code>tlim</code>	Modifications of the time X-axis limits. Vector with two values in seconds.
<code>trel</code>	Time X-axis with a relative scale when <code>tlim</code> is not null, <i>i.e.</i> relative to <code>wave</code> .
<code>mag.time</code>	Optional argument for magnifying the time coordinates. This is sometimes desired for small sound windows (<i>e.g.</i> less than 1 s), in which the time coordinates will be on a different scale than that of frequency coordinates. In those cases, it is recommended to include <code>mag.time=10</code> or <code>mag.time=100</code> , depending on length of sound window. By default, no magnification is performed (<i>i.e.</i> <code>mag.time=1</code>).
<code>EQ</code>	A vector of desired energy quantiles. By default, <code>SoundShape</code> computes 7 quantiles. When dealing with narrow banded calls, any additional values could cause an error in subsequent steps of the analysis.
<code>wav.at</code>	Filepath to the folder where wave files are stored. Should be presented between quotation marks. By default, <code>SoundShape</code> will use current working directory.
<code>store.at</code>	Filepath to the folder where <code>tps</code> file and spectrogram plots will be stored. Should be presented between quotation marks. By default, <code>SoundShape</code> will use current working directory.
<code>plot.exp</code>	A logical, if <code>TRUE</code> , <code>SoundShape</code> will export a file for each wave file in the indicated folder. Plots consist in spectrograms with longest relative contour and semilandmarks.
<code>plot.as</code>	Desired format for image files when <code>plot.exp=TRUE</code> . <code>"jpeg"</code> will generate compressed images for quick inspection of semilandmarks, whereas <code>"tiff"</code> or <code>"tif"</code> will generate uncompressed high resolution images that can be edited and used for publication.
<code>TPS.file</code>	Desired name for <code>tps</code> file containing semilandmark coordinates. Should be presented between quotation marks. By default = <code>NULL</code> , which prevents <code>SoundShape</code> from creating a <code>tps</code> file.

```

#~~~~~#
#                               SoundShape function:                               #
#   Computing semilandmarks from a curve of relative amplitude   #
#~~~~~#

SoundShape <- function(f=44100, wl=512, ovlp=50, dBlevel=25,
  flim=c(0, 6), tlim=c(0, 4), trel=c(0, 4), mag.time=1,
  wav.at = getwd(), store.at = getwd(),
  EQ= c(0.05, 0.15, 0.3, 0.5, 0.7, 0.85, 0.95),
  plot.exp=TRUE, plot.as="jpeg", TPS.file=NULL)

{
  # Require packages
  if(!library("seewave", logical.return=T)){install.packages(
"seewave")}
  if(!library("tuneR", logical.return=T)){install.packages(
"tuneR")}
  if(!library("abind", logical.return=T)){install.packages(
"abind")}
  require("seewave")
  require("tuneR")
  require("abind")

  # Function to compute zero-crossing for a curve of relative
  amplitude
  ZC <- function(x) {
    sign <- ifelse(x >= 0, 1, -1)
    res <- abs(diff(sign))
    return(res)
  }

  # Function to acquire frequency values for each energy
  quantiles
  frequencies <- function(spec, f=NULL, mel=FALSE, flim=NULL)
  {
    fhz <- f
    if (!is.null(f) & mel) {f <- 2 * mel(f/2)}
    if (is.null(f)) {
      if (is.vector(spec))

```

```

    stop("'f' is missing")
  else if (is.matrix(spec))
    f <- spec[nrow(spec),1]*2000*nrow(spec)/(nrow(spec) - 1)
  }
  if (is.matrix(spec)) {
    freq <- spec[, 1]
    freq = freq * 1000
    spec <- spec[, 2]
  }
  L <- length(spec)
  wl <- L * 2
  if (any(spec < 0))
    stop("The frequency spectrum to be analysed should not be
in dB")
  if (!is.null(flim)) {
    if (flim[1] < 0 || flim[2] > fhz/2)
      stop("'flim' should range between 0 and f/2")
    if (mel)
      flim <- mel(flim * 1000)/1000
  }
  else {flim <- c(0, f/2000)}

  spec <- spec[(flim[1] * 1000 * wl/f):(flim[2]* 1000 * wl/f)]
  amp <- spec/sum(spec)
  cumamp <- cumsum(amp)
  freq <- seq(from = flim[1] * 1000, to = flim[2] * 1000,
length.out = length(spec))

  for(i in EQ) # Frequency values according to desired energy
quantiles
  { if(!exists("Qlist")) {Qlist <- data.frame()}
    if(exists("Qlist")) {
      Qtemp <- data.frame("Quantiles"=paste("EQ-", i, sep=""),
"Frequency values" = freq[length(cumamp[cumamp <= i]) + 1])
      Qlist <- rbind(Qlist, Qtemp)
      rm(Qtemp)
    }
  }
}

```

```

    results <- Qlist
  }

  # Create TPS file before analysis
  if(!is.null(TPS.file)){
    TPS.path <- paste(store.at, "/", TPS.file, ".tps", sep="")
    file.create(TPS.path, showWarnings = TRUE)
  }

  # Zero crossing for each file on a given folder
  for(file in list.files(wav.at, pattern = ".wav"))
  {
    Wav <- readWave(paste(wav.at, "/", file, sep=""))

    # Store spectrogram as R object
    s <- spectro(Wav, f=f, wl=wl, ovlp=ovlp, flim=flim,
tlim=tlim, plot=FALSE)

    # Selection of longest relative amplitude contour
    con <- contourLines(x=s$time, y=s$freq, z=t(s$amp),
levels=seq(-dBlevel, -dBlevel, 1))
    n <- numeric(length(con))
    for(i in 1:length(con)) n[i] <- length(con[[i]]$x)
    n.max <- which.max(n)
    con <- con[[n.max]]

    # Initial and final time values within relative amplitude
    contour
    t.beg <- min(con$x)
    t.end <- max(con$x)

    # Minimum and maximum frequency values within relative
    amplitude contour
    f.min <- min(con$y) # Minimum frequency
    t.min <- con$x[con$y==f.min] # Time value associated

    f.max <- max(con$y) # Maximum frequency
    t.max <- con$x[con$y==f.max] # Time value associated
  }
}

```

```

# Power spectrum calculated within maximum and minimum
frequency values, and initial and final time values within the
longest contour
pwr.spec <- spec(Wav, f=f, wl=wl, from=t.beg, to=t.end,
plot=F)
fvalues <- frequencies(pwr.spec, f=f, flim=c(f.min, f.max))
fvalues$Frequency.values <- (fvalues$Frequency.values)/1000
# Convert values to kHz

# Acquire zero crossing between calculated frequency
quantiles and relative amplitude contour
for(i in seq_along(fvalues$Frequency.values)) {
  zc.res <- ZC(con$y - fvalues$Frequency.values[i])
  t.pos <- which(zc.res==2)

  if(!exists("t1") || !exists("t2")){
    t1 <- vector(mode="integer",
length=length(fvalues$Frequency.values))
    t2 <- vector(mode="integer",
length=length(fvalues$Frequency.values))
  }

  if(exists("t1") | exists("t2")){
    t1[i] <- min(con$x[t.pos])
    t2[i] <- max(con$x[t.pos])
  }
}

# Coordinates for plotting
SM1 <- data.frame(cbind(fvalues, "Time"= t1))
SM2 <- data.frame(cbind(fvalues, "Time"= t2))

# Re-order SM1 so that points plotting will be anticlockwise
SM1 <- SM1[order(SM1[,2], decreasing=T), ]

# Store SM-min and SM-max in a data.frame
SM.min <- data.frame("Quantiles" = "EQ-min",

```

```

"Frequency.values"= f.min, "Time"= t.min)
  SM.max <- data.frame("Quantiles" = "EQ-max",
"Frequency.values"= f.max, "Time"= t.max)

  # Semilandmark numbers
  SM.numbers <- as.factor(1:((length(EQ)*2) + 2)) # Twice the
length of EQ plus the SM-min and SM-max

  # Complete data.frame with coordinates of x and y
  SM <- cbind(SM.numbers,data.frame(rbind(SM.max, SM1, SM.min,
SM2)))

  # Magnify time coordinates
  SM[,4] <- SM[,4]*mag.time

  # Store semilandmark coordinates as array of matrices
  SM.mtx <- as.matrix(SM[,4:3])
  ifelse(!exists('coordSM'),
    coordSM <- array(data=SM.mtx, dim=c(dim(SM.mtx), 1)),
    coordSM <- abind::abind(coordSM, SM.mtx, along=3))

  # Plot over spectrogram
  if(plot.exp == TRUE){
    if(plot.as == "jpeg"){jpeg(width =5000,height = 3500,
units = "px", res = 500,
    filename=paste(store.at,"/", sub(".wav","",file),
".jpg", sep=""))} # compressed images
    if(plot.as=="tiff"|plot.as=="tif"){tiff(width=5000,
height=3500, units="px", res=500, filename=paste(store.at, "/",
sub(".wav", "", file), ".tif", sep=""))} # uncompressed images
    spectro(Wav, f=f, wl=wl, ovlp=ovlp,osc=F, scale=F, grid=F,
    main= sub(".wav", "", file), trel=trel, flim=flim,
tlim=tlim)

    polygon(x=con$x, y=con$y, lwd=1.5)
    points(x=(SM$Time)/mag.time,y=SM$Frequency.values, pch=21,
col="black", bg="red", cex=1.1)
    text(x=(SM$Time)/mag.time, y=SM$Frequency.values, cex=0.7,
labels=SM$SM.numbers, pos=c(3, rep(2, dim(SM1)[1]), 1,

```

```

rep(4, dim(SM2)[1])) # Sm numbers images

    dev.off()
}

# Export semilandmark coordinates as TPS file
if(!is.null(TPS.file)){
  lmline <- paste("LM=", dim(SM)[1], sep="")
  write(lmline, TPS.path, append = TRUE)
  write.table(SM[,4:3], TPS.path, col.names = FALSE,
row.names = FALSE, append = TRUE)
  idline <- paste("ID=", sub(".wav", "", file), sep="")
  write(idline, TPS.path, append = TRUE)
  write("", TPS.path, append = TRUE)
  rm(lmline, idline, SM)
}

}

# Assign names to arrays' third dimension
dimnames(coordSM)[[3]] <- sub(".wav", "", list.files(wav.at,
pattern = ".wav"))
results <- coordSM
}

```

Using SoundShape function

In order to illustrate how to use `SoundShape` function, we selected a representative note of each *Physalaemus* species and stored them on a single folder. For illustrative purposes, we'll define this folder as our working directory, but store the spectrogram plots and the `tps` file elsewhere by specifying a different path for the outputs.

R code: The working directory is set using the `setwd` function. We call `dir(pattern=".wav")` to visualize wave files stored on the selected folder. Two alternative ways to use `SoundShape` function are employed. We first use the function controlling all of its arguments, which are basically the same as in the default of `SoundShape`. Alternatively, we can focus on the arguments we wish to control, and

SoundShape will run its default settings. The second alternative is much simpler and recommended to avoid typing errors.

```
# Source SoundShape function into R's environment
source("insert.filepath.here/SoundShape.R")

# Define working directory where wave files are stored
setwd("C:/R/SoundShape-Sample")
dir(pattern=".wav")

## [1] "P-aguirrei_Linhares-ES_FNJV-0031776.wav"
## [2] "P-albifrons_Maracas-BA_FNJV-0031943.wav"
## [3] "P-albonotatus_Caceres-MT_FNJV-0031171.wav"
## [4] "P-biligonigerus_Osorio-RS_FNJV-0031682.wav"
## [5] "P-bokermanni_Santo-Andre-SP_FNJV-0031184.wav"
## [6] "P-camacan_Igrapiuna-BA_FNJV-0034021.wav"
## [7] "P-centralis_Barra-do-Bugres-MT_FNJV-0031235.wav"
## [8] "P-cicada_Patos-PB_FNJV-0031191.wav"
## [9] "P-crombiei_Aracruz-ES_FNJV-0032255.wav"
## [10] "P-cuvieri_Sao-Jose-Campos-SP_CBUFG-196.wav"
## [11] "P-deimaticus_Serra-do-Cipo-MG_CBUFG-173.wav"
## [12] "P-erythros_Ouro-Preto-MG_CBUFG-152.wav"
## [13] "P-evangelistai_Santana-do-Riacho-MG_FNJV-0031636.wav"
## [14] "P-feioi_Vicosa-MG_FNJV-0031559.wav"
## [15] "P-fischeri_Mantecal_Venezuela_FNJV-0031232.wav"
## [16] "P-gracilis_Porto-Alegre-RS_FNJV0031609.wav"
## [17] "P-kroyeri_Ilheus-BA_FNJV-0032047.wav"
## [18] "P-lisei_Sao-Francisco-Paula-RS_FNJV-0031645.wav"
## [19] "P-maculiventris_Salesopolis-SP_FNJV-0033756.wav"
## [20] "P-marmoratus_Luis-Antonio-SP_FNJV-0004592.wav"
## [21] "P-moreirae_Salesopolis-SP_FNJV-0033755.wav"
## [22] "P-nanus_Rancho-Queimado-SC_FNJV-0031597.wav"
## [23] "P-nattereri_Caceres-MT_FNJV-0031253.wav"
## [24] "P-obtectus_Linhares-ES_FNJV-0031774.wav"
## [25] "P-olfersii_Sao-Jose-do-Barreiro-SP_FNJV-0031263.wav"
## [26] "P-riograndensis_Eldorado-do-Sul-RS_FNJV-0031643.wav"
## [27] "P-rupestris_Serra-do-Ibitipoca-MG_CBUFG-875.wav"
## [28] "P-signifer_Itaguaí-RJ_FNJV-0031273.wav"
## [29] "P-soaresi_Itaguaí-RJ_FNJV-0031272.wav"
## [30] "P-spiniger_Caraguatatuba-SP_FNJV-0031570.wav"
```

```

# Compute semilandmarks for each wave file at working
directory and store them elsewhere

SS.sample <- SoundShape(f=44100, wl=512, ovlp=70, dBlevel=25,
tlim = c(0, 4), trel=c(0, 4), flim=c(0, 6),
EQ= c(0.05, 0.15, 0.3, 0.5, 0.7, 0.85, 0.95), wav.at = getwd(),
store.at = "C:/R/SoundShape-Sample/Output", plot.exp=TRUE,
plot.as="jpeg", TPS.file = "SS_sample")

# Simpler way to call SoundShape function. Arguments omitted
imply that default will be used

SS.sample <- SoundShape(TPS.file = "SS_sample",
store.at="C:/R/SoundShape-Sample/Output")

```

R code: We can explore the contents of the `SoundShape` object by calling a few base functions, such as the `class` of the object, its dimensions `dim` (the result of `dim` is in the form (rows, columns, elements)), and the `class` of each element in the array. Ultimately, we can verify the semilandmark coordinates by calling the `SoundShape` object created.

```

# Verify contents of object created
class(SS.sample)

## [1] "array"

dim(SS.sample)

## [1] 16  2 30

class(SS.sample[, , 1])

## [1] "matrix"

# Visualize semilandmark coordinates from first two elements
SS.sample[, , 1:2]

## , , P-aguirrei_Linhares-ES_FNJV-0031776
##
##      Time Frequency.values
## 1  0.6462882      2.877923
## 7  0.6393013      2.863333
## 6  0.6358079      2.826857
## 5  0.6305443      2.724725
## 4  0.6672489      2.560583

```

```
## 3 0.6812227 2.403737
## 2 0.7056769 2.119225
## 11 0.7545852 1.896722
## 12 0.9013100 1.645039
## 13 0.8593886 1.896722
## 21 0.7441048 2.119225
## 31 0.7106386 2.403737
## 41 0.6850439 2.560583
## 51 0.6742358 2.724725
## 61 0.6602620 2.826857
## 71 0.6572068 2.863333
##

## , , P-albifrons_Maracas-BA_FNJV-0031943
##
##      Time Frequency.values
## 1 0.5414847 2.9297210
## 7 0.4471616 2.5016796
## 6 0.4974180 2.3434121
## 5 0.4331878 1.9657285
## 4 0.4361397 1.6959545
## 3 0.4366812 1.3434498
## 2 0.4262009 0.8218866
## 11 0.4436681 0.4046361
## 12 0.6567686 0.1600410
## 13 0.6707424 0.4046361
## 21 0.6777293 0.8218866
## 31 0.6879394 1.3434498
## 41 0.6851355 1.6959545
## 51 0.6812227 1.9657285
## 61 0.6867349 2.3434121
## 71 0.6799961 2.5016796
```

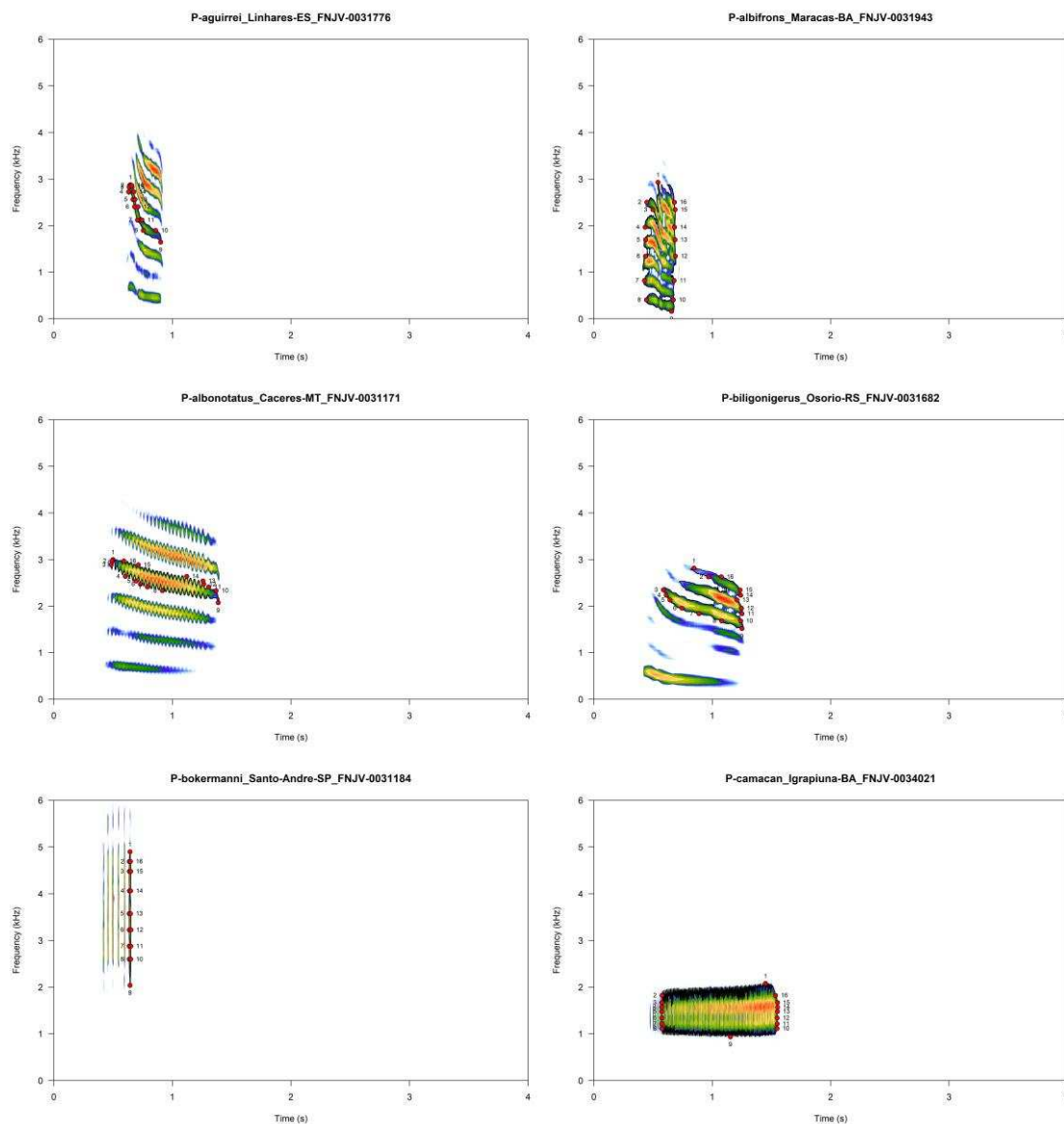
By assigning the result of `SoundShape` to an R object, semilandmark coordinates are stored as a 3D array, which is a special type of `list` that can be thought as a filing cabinet, in which the `array` is the cabinet and each element is an archive. This special list will be used by `geomorph` package in the subsequent steps of the analysis. In our sample, `SS.sample` is an array composed of 30 elements (*i.e.* 30 species of

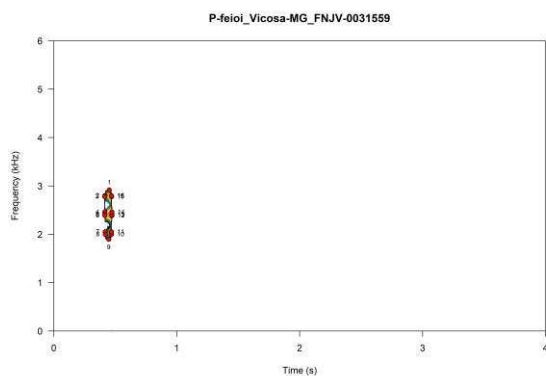
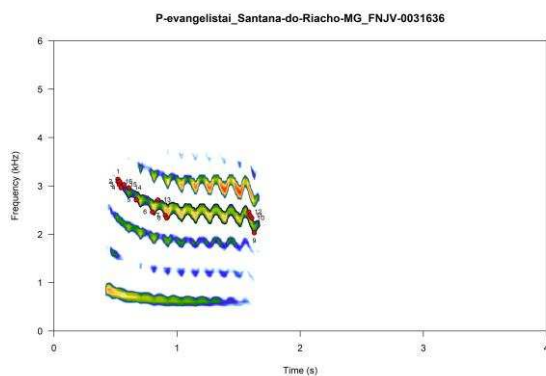
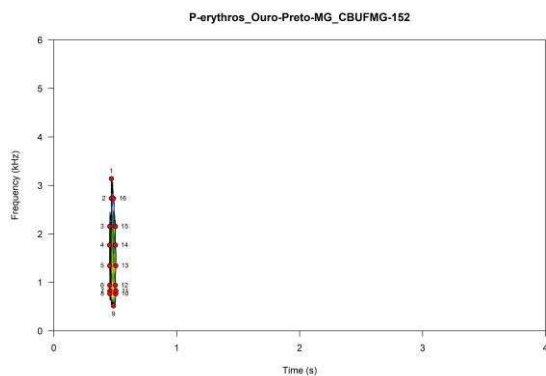
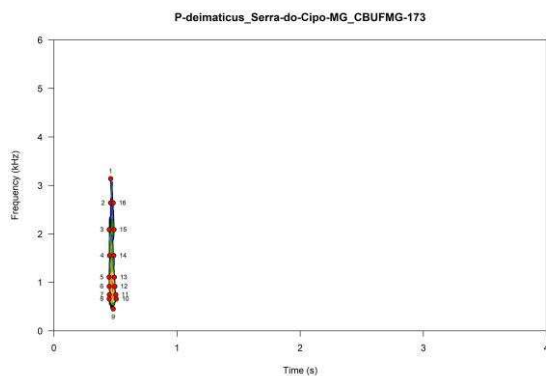
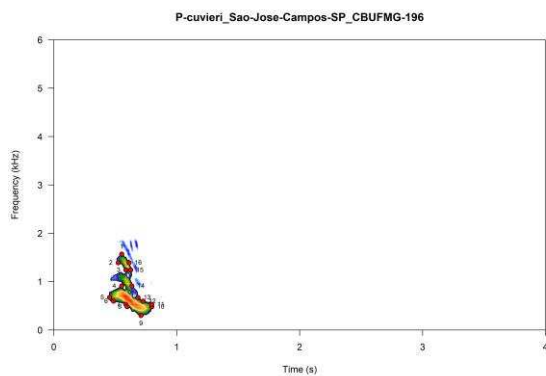
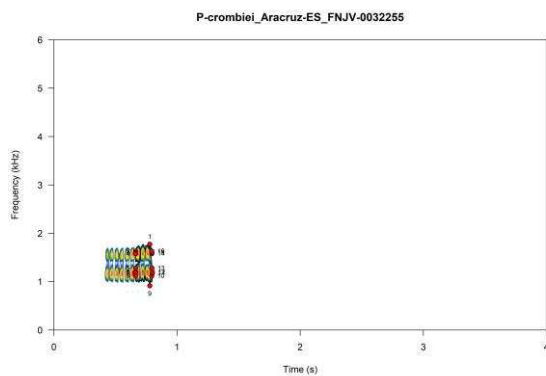
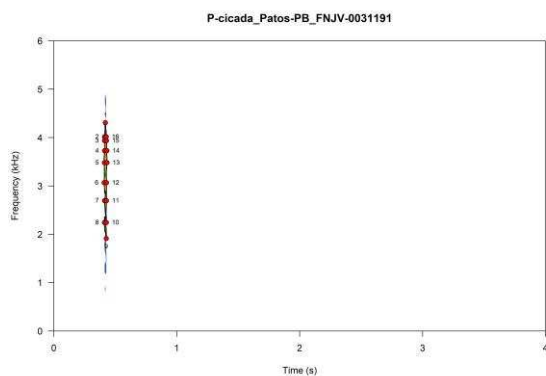
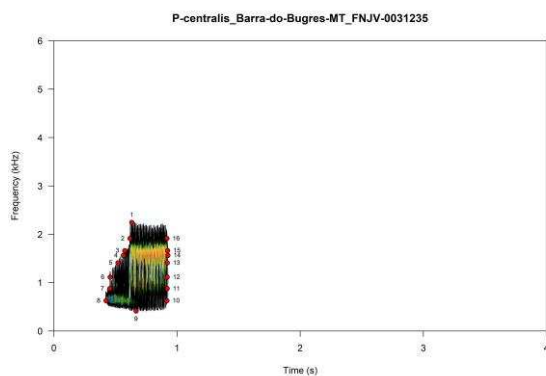
Physalaemus), each one is a matrix with 16 rows and 2 columns (i.e. X and Y coordinates of 16 semilandmarks).

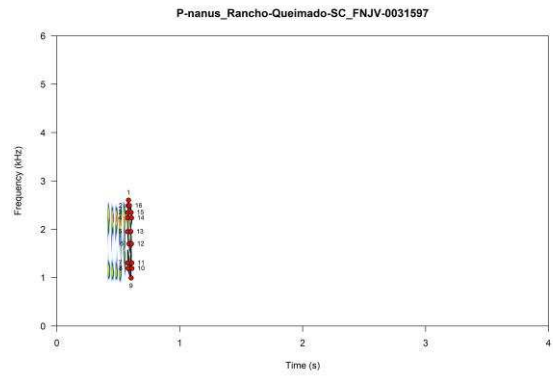
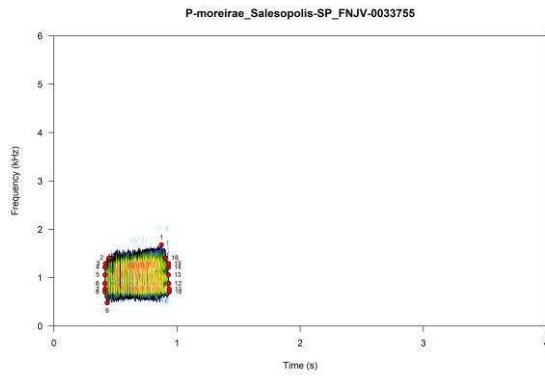
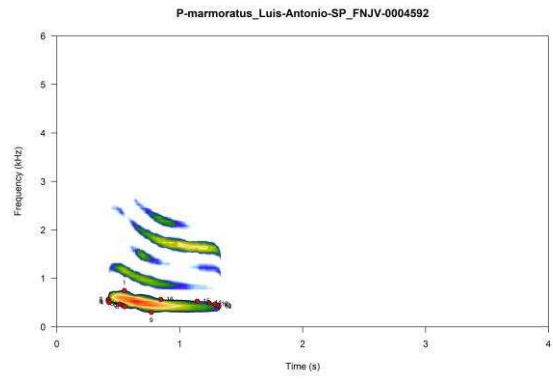
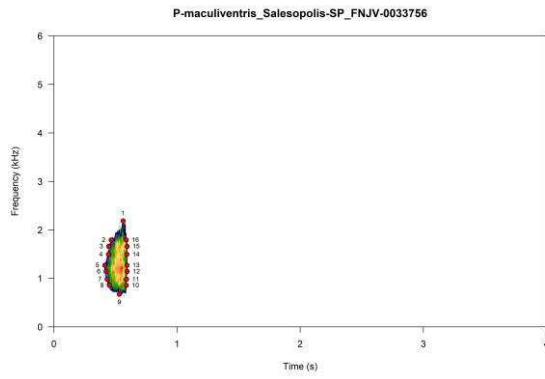
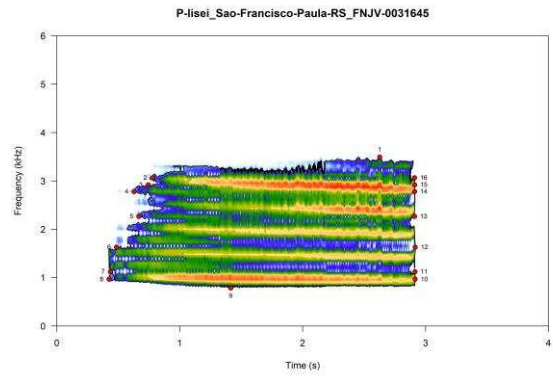
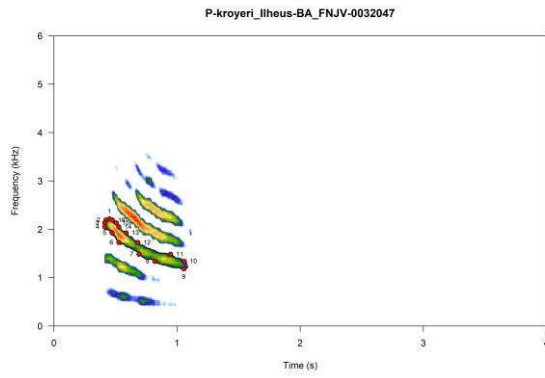
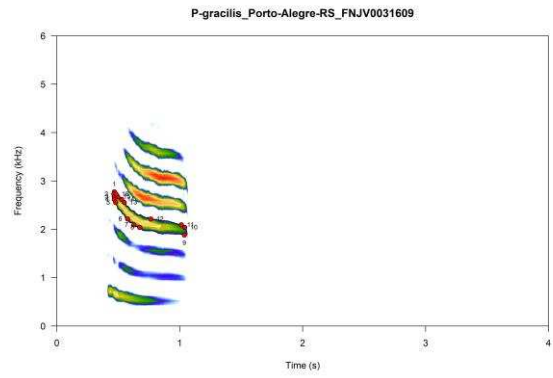
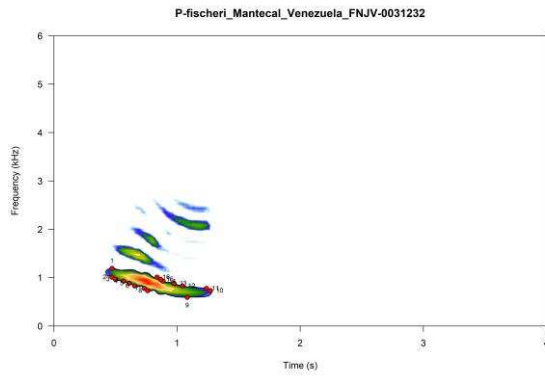
Ultimately, we can now check how semilandmarks were plotted in each species (Figure 19).

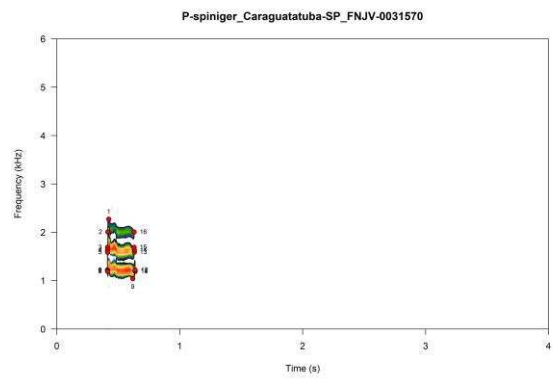
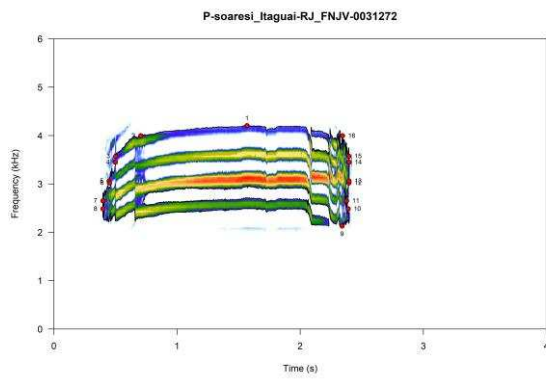
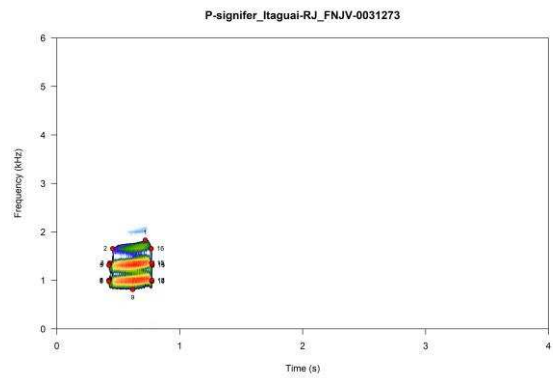
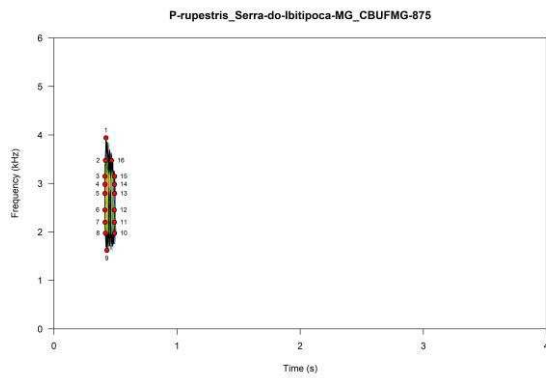
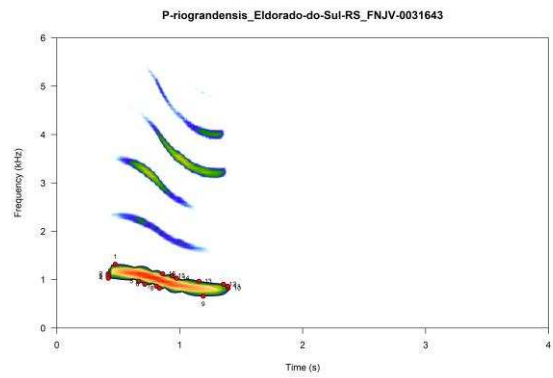
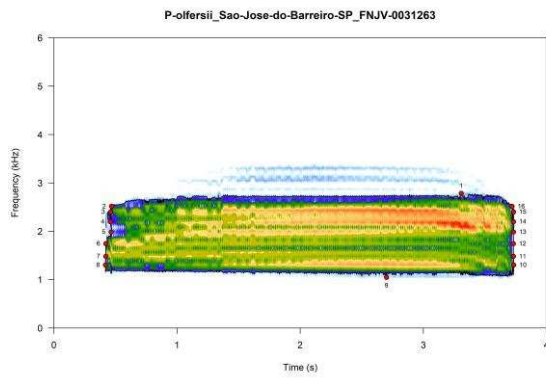
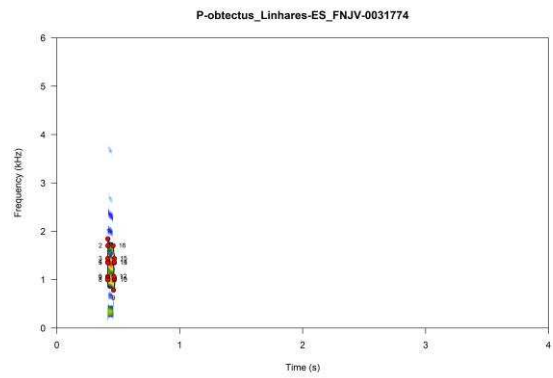
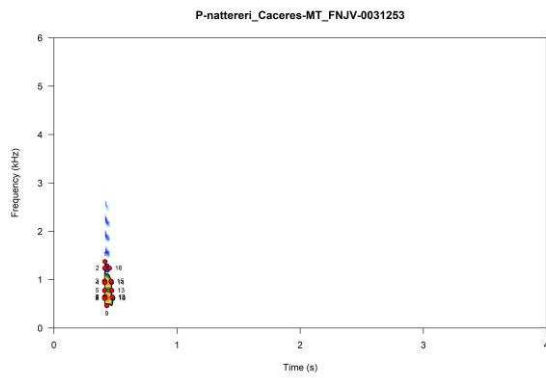
Figure 19: (Displayed below) Spectrogram views of the advertisement calls of 30 *Physalaemus* species with complete set of 16 semilandmarks acquired as cross correlation between longest curve of relative amplitude at -25 dB, and frequency values representing energy quantiles within the contour. Each species had a representative call/note selected previously and spectrograms were generated automatically by SoundShape function.

```
# Display spectrograms from folder
include_graphics(list.files("C:/R/SoundShape-Sample/Output",
pattern=".jpg", full.names = TRUE))
```









Before we finally move into the Geometric Morphometric steps of the analysis, we will explore a different method for describing the shape of sound using semilandmarks. So far, we have dealt with a two dimensional spectrogram shape by acquiring semilandmarks from a curve of relative amplitude. Now we focus on the *eigensound* approach, which is described by MacLeod et al. (2013) as a useful taxonomic tool applied to bat echolocation calls.

Herein, we will go through the method without the details provided by MacLeod et al. (2013). Our focus is to provide a new function `eigensound` that works similarly as the `SoundShape`, enabling future studies to employ both methods with ease.

First, we must take a deeper look into the graphical representation of sound.

***Eigensound* 3D analysis: Semilandmarks from the three dimensional surface of sound**

Understanding the matrix of amplitude values

The *eigensound analysis*, as described by MacLeod et al. (2013), is a method derived from the Geometric Morphometrics and employed a morphometric empirical study of bat echolocation calls. The method considers spectrograms as complex three dimensional surfaces, in which a call sampling grid is applied over the sound surface. Each node in the grid bear an amplitude value whose time and frequency coordinates (*i.e.* X and Y-axis) are identical throughout the grid, meaning that only the amplitude values vary. Therefore, each node in the grid is considered a topologically homologous semilandmark described by a matrix of three dimensional coordinates, in which x = time axis, y = frequency axis, and z = amplitude axis.

The matrix of amplitude values is generated by a series of Fast Fourier Transformations (FFT), which is an algorithm that decompose the time per amplitude data (*i.e.* oscillogram) into a set of frequency values with associated amplitudes (*i.e.* power spectrum; Charif et al., 2010, Köhler et al., 2017). A spectrogram is, essentially, a series of FFT transformations for each discrete time interval and overlapped in the time axis, thus forming a matrix in which the rows are represented by the frequencies, columns are the time values, and each cell represents an amplitude value. By assigning a color pallete to these amplitude values, we achieve a 2D spectrogram. If, on the other hand, we assign an axis (*i.e.* z-axis) for the amplitude, we can then create the 3D spectrogram (Figure 3, 4, 5 and 6).

Computationally, we can dissect the spectrogram output by assigning the result of `spectro` to an R object. To do so, we will continue to work with the edited calls of *Physalaemus centralis* and *P. kroyeri* before encompassing the remaining species of *Physalaemus*.

R code: In this exploratory phase, we will read the same wave files used previously and then create and assign the spectrogram plot to an R object without actually plotting it. At this point, it does not matter which wave file will be used to create the `spectro` object as it will be used only to explore its contents.

```

# Read wave files
P.eph <- readWave("FNJV-0031235_P-centralis_BB1_003.wav") #
P. centralis

P.kro <- readWave("FNJV-0032047_P-kroyeri_I11_002.wav")
# P. kroyeri

# Assign spectrogram output to an object. There is no need to
plot the spectrogram yet
s.eph <- spectro(P.eph, f=44100, plot=F, tlim=c(0,1), flim=c(0,4))

```

R code: The code below is used to verify the class of the `spectro` object created. Then we use `lapply` to check the class and the length of each element in the list of `s.eph`. The matrix dimensions `dim` has to be checked separately as `length` does not apply to matrix objects.

```

# Verify the contents of the new object created
class(s.eph)

## [1] "list"

lapply(s.eph, class)

## $time
## [1] "numeric"
##
## $freq
## [1] "numeric"
##
## $amp
## [1] "matrix"

# Dimensions of each element in the list
lapply(s.eph[1:2], length)

## $time
## [1] 86
##
## $freq
## [1] 47

```

```
dim(s.eph$amp)

## [1] 47 86
```

The spectrogram output is a list of three elements: two numeric vectors `$time` and `$freq`, representing the X and Y coordinates, respectively, and the matrix of amplitude values `$amp`, whose dimensions match the length of `$freq` at the number of rows, and the length of `$time` at the number of columns. Therefore, we can focus our analysis on the amplitude matrix, as it encompass all elements in a spectrogram.

Also worth noting that the amplitude matrix will always have the same dimensions for a given set of spectrogram parameters, regardless of the sound wave used to produce the spectrogram.

R code: Assign each eave file to a different `spectro` object similarly as the last code chunk. Use `identical` function to verify if the dimensions and the coordinates of `$amp` are the same.

```
# Create a spectrogram object for two distinct sound waves
s.eph <- spectro(P.eph, plot=F, f=44100, wl=512, ovlp=70,
tlim=c(0,1), flim=c(0,4))
s.kro <- spectro(P.kro, plot=F, f=44100, wl=512, ovlp=70,
tlim=c(0,1), flim=c(0,4))

# Verify if the dimensions of both amplitude matrixes are identical
identical(dim(s.eph$amp), dim(s.kro$amp))

## [1] TRUE

# Verify if x and y coordinates are identical
identical(s.eph$time, s.kro$time)

## [1] TRUE

identical(s.eph$freq, s.kro$freq)

## [1] TRUE
```

Intuitively, if one wish to change the spectrogram limits, the dimensions of `$amp` will change accordingly.

R code: Create new `spectro` objects for a new set of spectrogram parameters. Changes in the spectrogram's X-axis (*i.e.* time axis) are made using `tlim` argument, whereas changes in the Y-axis (*i.e.* frequency axis) are controlled by `flim`.

```
# Changing spectrogram limits

# Increase time (x axis)
s.eph <- spectro(P.eph, plot=F, f=44100, tlim=c(0,2),
flim=c(0,4)) # Time ranging from 0 to 2 s

dim(s.eph$amp)

## [1] 47 172

s.eph <- spectro(P.eph, plot=F, f=44100, tlim=c(0,4),
flim=c(0,4)) # Time ranging from 0 to 4 s

dim(s.eph$amp)

## [1] 47 344

# Increase frequency (y axis)
s.eph <- spectro(P.eph, plot=F, f=44100, tlim=c(0,4),
flim=c(0,6)) # Frequency ranging from 0 to 6 kHz

dim(s.eph$amp)

## [1] 70 344
```

Additionally, we can alter the FFT itself by changing the spectrograms' window length `wl`, which also influences the `$amp` matrix dimensions. More importantly, changes in the `wl` result in changes in the spectrogram output (Figure 20, 21 and 22), thus implying that the `wl` can have a direct impact on the results from both `eigensound` and `SoundShape` approaches.

Note: The code chunk below focus on how the window length `wl` impact the amplitude matrix dimensions and the spectrogram output. Therefore, total duration is reduced to 1 s and frequency ranges restricted from 0 to 4 kHz, thus reducing the dimensions of amplitude matrix `s.eph` and prioritizing the better visualization of spectrogram output.

R code: Once again we assign the result of `spectro` to an object, however we now allow the spectrogram to be plotted by calling the argument `plot=TRUE`. This is done to better illustrate the output changes when the argument window length `wl` is changed. Note that we first call the function `par` to enable both spectrograms to be plotted side by side.

```
# Changing window length (FFT)

# Default window length (wl=512)
par(mfrow=c(1,2)) # Plot spectrograms side by side
s.eph <- spectro(P.eph, plot=T, grid=F, scale=F, f=44100,
tlim=c(0,1), flim=c(0,4), wl=512,
main=expression(italic('Physalaemus centralis'))))
s.kro <- spectro(P.kro, plot=T, grid=F, scale=F, f=44100,
tlim=c(0,1), flim=c(0,4), wl=512,
main=expression(italic('Physalaemus kroyeri')))
```

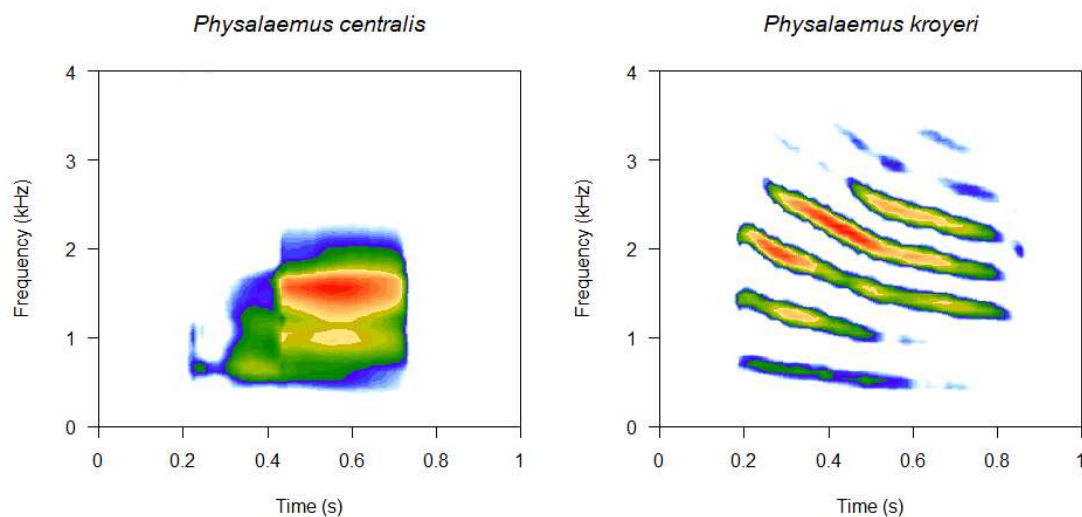


Figure 20: Spectrogram view of the advertisement calls of *Physalaemus centralis* (left) and *P. kroyeri* (right) computed with default medium-sized window length `wl=512`.

```
# Dimensions of amplitude matrix with default window length
(wl=512)
dim(s.eph$amp)

## [1] 47 86
```

```
# Small window (wl=256)
par(mfrow=c(1,2))
s.eph <- spectro(P.eph, grid=F, scale=F, f=44100, tlim=c(0,1),
flim=c(0,4), wl=256,
main=expression(italic('Physalaemus centralis'))
s.kro <- spectro(P.kro, grid=F, scale=F, f=44100, tlim=c(0,1),
flim=c(0,4), wl=256,
main=expression(italic('Physalaemus kroyeri'))
```

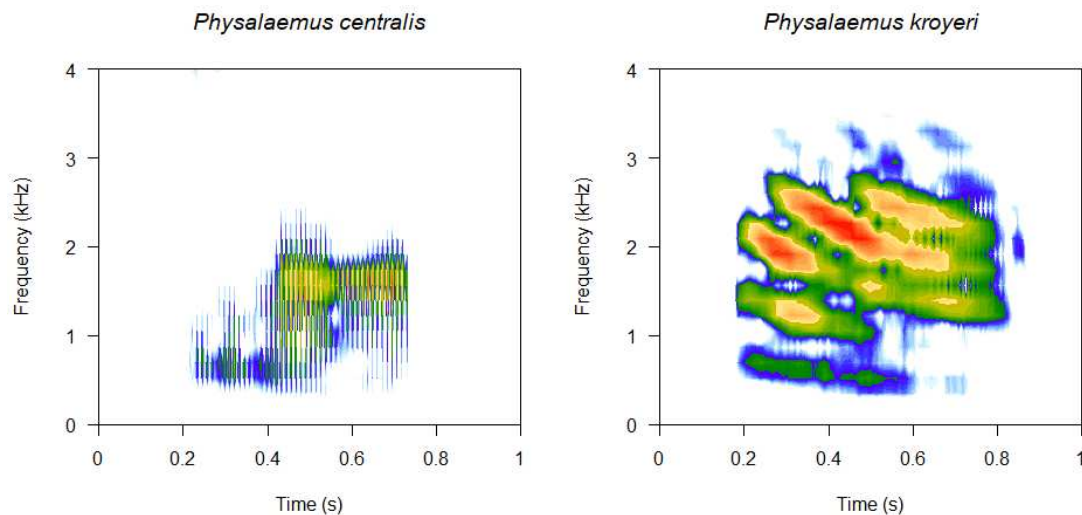


Figure 21: Spectrogram view of the advertisement calls of *Physalaemus centralis* (left) and *P. kroyeri* (right) computed with small-sized window length $wl=256$.

```
# Dimensions of amplitude matrix using small window (wl=256)
dim(s.eph$amp)

## [1] 24 172

# Large window (wl=1024)
par(mfrow=c(1,2))
s.eph <- spectro(P.eph, grid=F, scale=F, f=44100, tlim=c(0,1),
flim=c(0,4), wl=1024,
main=expression(italic('Physalaemus centralis'))
s.kro <- spectro(P.kro, grid=F, scale=F, f=44100, tlim=c(0,1),
flim=c(0,4), wl=1024,
main=expression(italic('Physalaemus kroyeri'))
```

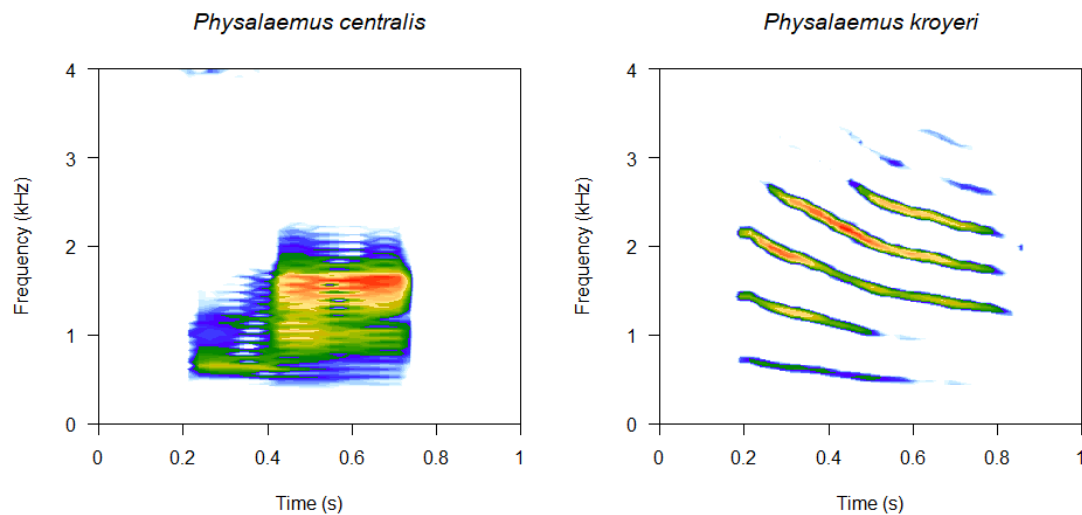


Figure 22: Spectrogram view of the advertisement calls of *Physalaemus centralis* (left) and *P. kroyeri* (right) computed with default large-sized window length $wl=1024$.

```
# Dimensions of amplitude matrix using large window (wl=1024)
dim(s.eph$amp)

## [1] 93 43
```

As it can be seen from the spectrograms above, the window length has the potential to dramatically change the spectrogram output in the analysis.

By choosing a small window, each Fast Fourier Transform is computed from a short time interval, thus resulting in more columns in our matrix and an increased time resolution in the spectrogram (*e.g.* pulses more clearly defined). On the other hand, the smaller window imply on loss of frequency resolution (*e.g.* blurred harmonic structures) and less rows in our matrix.

Alternatively, the large window mean that the FFT will compute larger chunks of time, resulting in less columns in the amplitude matrix and poor time resolution in the spectrogram (*e.g.* pulses indistinguishable). As counterpart, the frequency values are emphasized (*e.g.* frequency harmonics clearly defined), which lead to more rows in the matrix.

Therefore, choosing the ideal window length is pivotal in any bioacoustical study, specially when dealing with the shape of sound. In most cases, the best option is to use an intermediate value (*i.e.* $wl=512$) that will result in an average spectrogram output (see Köhler et al. 2017).

Ultimately, we can increase spectrogram resolution by overlapping the windows. This is also a tradeoff option, as large values of overlap will increase data dimensionality and redundancy with little increase of resolution (Figure 23, 24 and 25).

R code: Overlap between windows can be controlled by the argument `ovlp`, which is given as percentage.

```
# Changing overlap between successive windows

# Little overlap between windows (ovlp=40%)
par(mfrow=c(1,2))
s.eph <- spectro(P.eph, grid=F, scale=F, f=44100, tlim=c(0,1),
flim=c(0,4), wl=512, ovlp=40,
main=expression(italic('Physalaemus centralis')))
s.kro <- spectro(P.kro, grid=F, scale=F, f=44100, tlim=c(0,1),
flim=c(0,4), wl=512, ovlp=40,
main=expression(italic('Physalaemus kroyeri')))
```

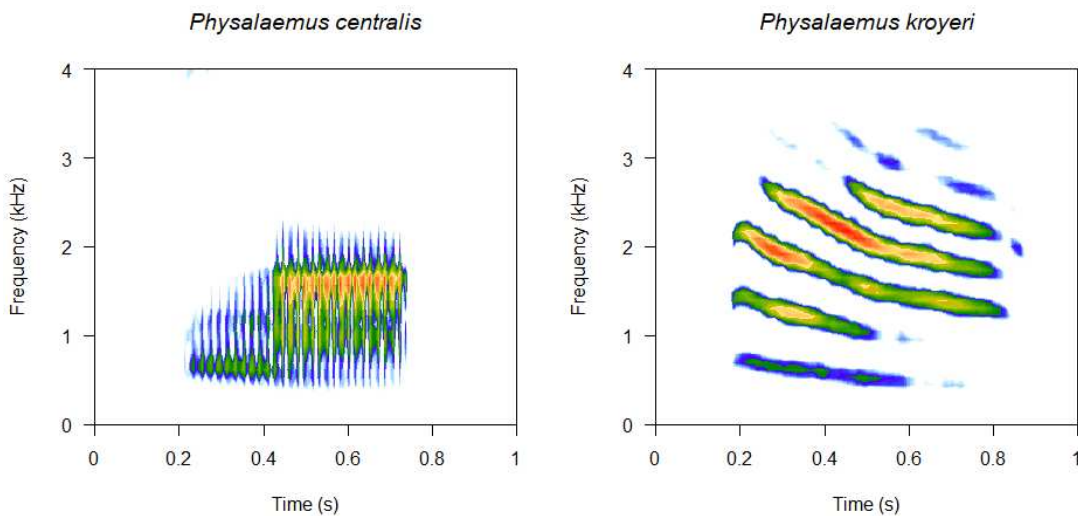


Figure 23: Spectrogram view of the advertisement calls of *Physalaemus centralis* (left) and *P. kroyeri* (right) computed with little overlap `ovlp=40` between successive windows.

```
# Dimensions of amplitude matrix when ovlp=40
dim(s.eph$amp)

## [1] 47 142

# Medium overlap between windows (ovlp=60%)
par(mfrow=c(1,2))
s.eph <- spectro(P.eph, grid=F, scale=F, f=44100, tlim=c(0,1),
flim=c(0,4), wl=512, ovlp=60,
```

```
main=expression(italic('Physalaemus centralis'))
s.kro <- spectro(P.kro, grid=F, scale=F, f=44100, tlim=c(0,1),
flim=c(0,4), wl=512, ovlp=60,
main=expression(italic('Physalaemus kroyeri')))
```

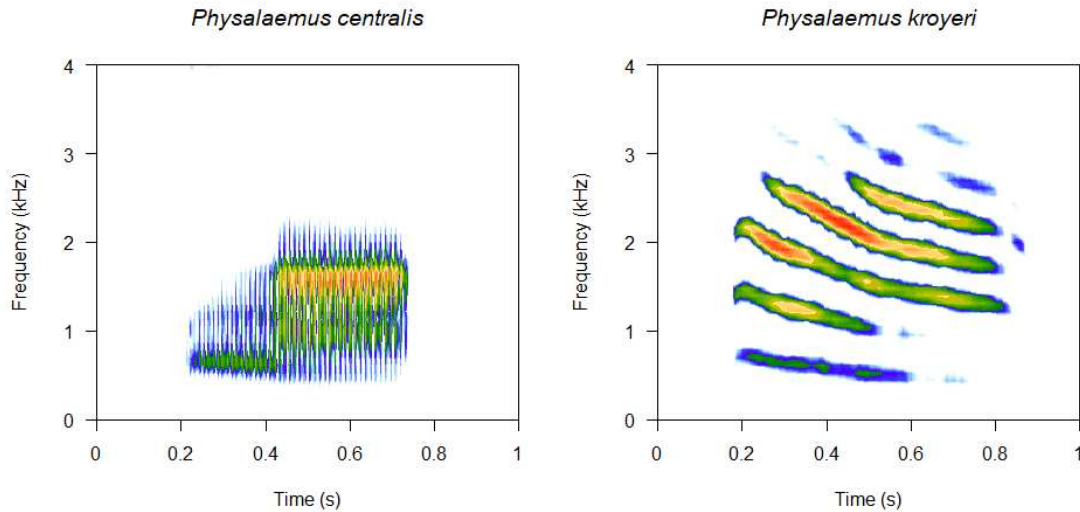


Figure 24: Spectrogram view of the advertisement calls of *Physalaemus centralis* (left) and *P. kroyeri* (right) computed with little overlap $ovlp=60$ between successive windows.

```
# Dimensions of amplitude matrix when ovlp=60
dim(s.eph$samp)

## [1] 47 213

# Increased overlap between windows (ovlp=80%)
par(mfrow=c(1,2))
s.eph <- spectro(P.eph, grid=F, scale=F, f=44100, tlim=c(0,1),
flim=c(0,4), wl=512, ovlp=80,
main=expression(italic('Physalaemus centralis'))
s.kro <- spectro(P.kro, grid=F, scale=F, f=44100, tlim=c(0,1),
flim=c(0,4), wl=512, ovlp=80,
main=expression(italic('Physalaemus kroyeri')))
```

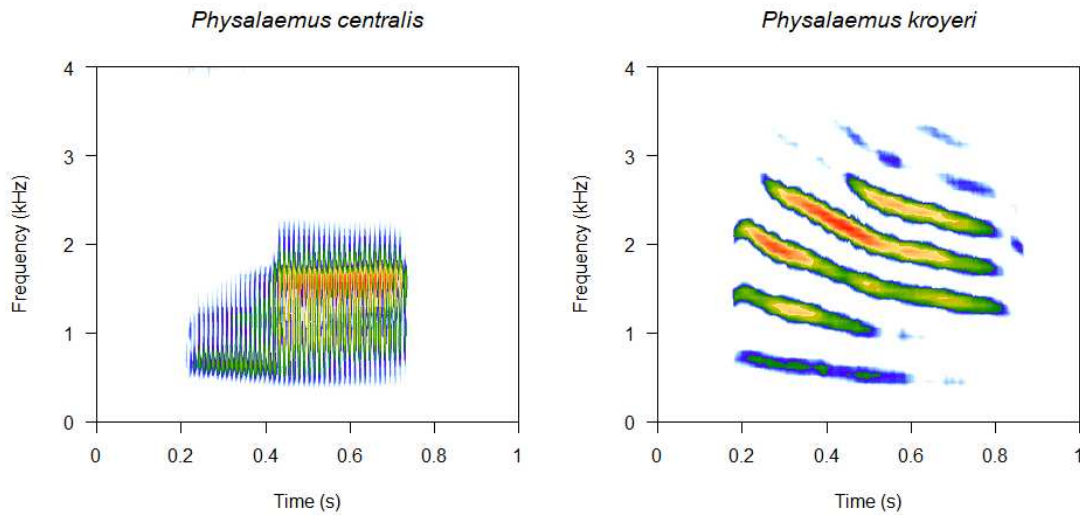


Figure 25: Spectrogram view of the advertisement calls of *Physalaemus centralis* (left) and *P. kroyeri* (right) computed with little overlap `ovlp=80` between successive windows.

```
# Dimensions of amplitude matrix when ovlp=80
dim(s.eph$amp)

## [1] 47 426
```

As long as spectrograms are calculated using the same parameters, the x and y coordinates will be the same for every sound window, meaning that each node (*i.e.* intersection between x and y coordinates) represents a topologically homologous semilandmark (MacLeod et al., 2013).

Reducing data dimensionality in the matrix of amplitude values

Similarly as the curve of relative amplitude from *SoundShape*, each matrix of amplitude values had its background amplitude reassigned at -25 dB, which reduces the influence from background noise. In addition, all calls were set to begin at time = 0, with the time X -axis set to range from 0 to 4 s, and frequency Y -axis from 0 to 6 kHz (Figure 26 and 27). Moreover, the sound windows were standardized with window length `wl=512`, overlap `ovlp=70` and sampling rate `f=44100`, reassuring that variation in each semilandmark is due to energy shifts within the call, and not to changes in their position within the sound window or in the window itself. These procedures are the same as described in MacLeod et al. (2013) and will be employed hereafter in both *eigensound* and *SoundShape* approaches.

Note: Although we could visualize the grid of amplitude coordinates embedded in our spectrogram, the grid itself is so dense that it is not feasible to visualize it.

R code: Read the wave files in which the calls are set to begin at time=0 and assign each spectrogram to an R object. Spectrograms are allowed to be plotted to enable visualization of notes at beginning of sound window.

```
# Spectrogram plot with call placed at time=0
eig.eph <- readWave("FNJV-0031235_P-centralis_eigen sound.wav")

s.eph <- spectro(eig.eph, grid=F, scale=F, f=44100, tlim=c(0,4), f
lim=c(0,6), wl=512, ovlp=70,
ain=expression(italic('Physalaemus centralis')))
```

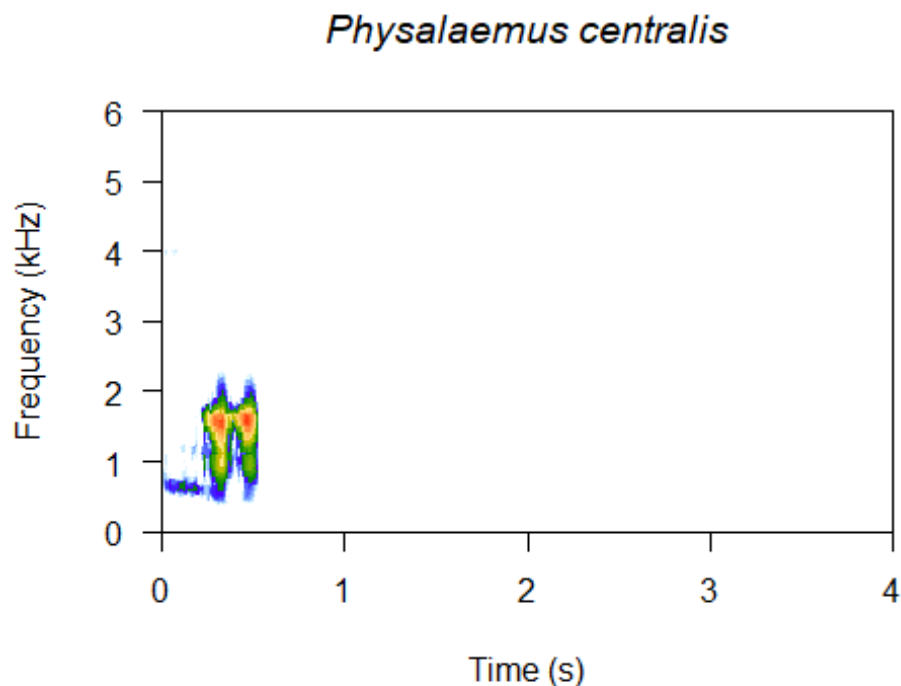


Figure 26: Spectrogram view of the advertisement call of *Physalaemus centralis* placed at time = 0 s.

```
# Spectrogram plot with call placed at time=0
eig.kro <- readWave("FNJV-0032047_P-kroyeri_eigen sound.wav")

s.kro <- spectro(eig.kro, grid=F, scale=F, f=44100, tlim=c(0,4),
flim=c(0,6), wl=512, ovlp=70,
main=expression(italic('Physalaemus kroyeri')))
```

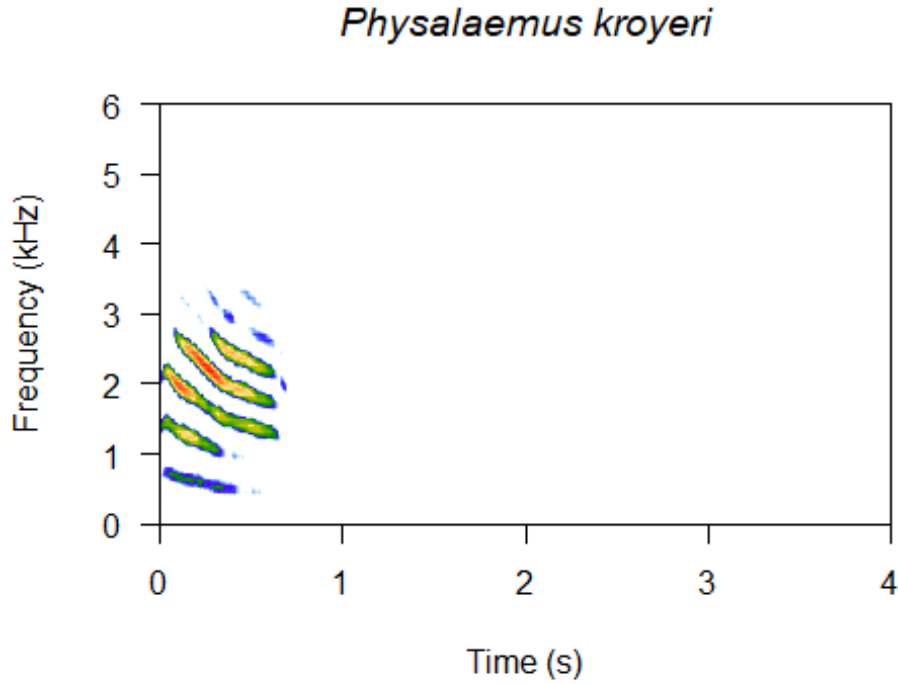


Figure 27: Spectrogram view of the advertisement call of *Physalaemus kroyeri* placed at time = 0 s.

A reasonable spectrogram resolution imply on a very large matrix of amplitude values. This is represented by a dense grid of x and y coordinates, which result on a large number of semilandmarks to compare. More specifically, by using the aforementioned spectrogram parameters, our matrix encompass 70 rows and 1146 columns, which lead to 80220 semilandmarks, each one representing a node from a very dense grid of amplitude coordinates. In other words, we are using 80220 variables to describe the surface of sound, the equivalent of 80220 columns in a dataset.

This very high data dimensionality can be reduced by creating a new sampling grid over the current grid of amplitude coordinates. Since MacLeod et al. (2013) dealt with short echolocation calls (*i.e.* sound window with total duration of 0.1023 s), they managed to create a rather small grid, with 30 (or 25) cells per side, which yielded the comparison of 900 semilandmarks. However our analysis encompassed species whose advertisement call reaches almost four seconds of a single note (*e.g.* *Physalaemus olfesi*), whereas others presented long calls composed of very short notes with average duration of 0.05 seconds (*e.g.* *P. cicada*, *P. erythros*, *P. deimaticus*) (see also Rocha & Romano *in prep*), meaning that the sound windows of our empirical data had to be much larger than the data used by MacLeod et al. (2013).

A large sound window also mean a large matrix of amplitude values, and thus 30 cells per side would be an oversimplification of the sound surface. Therefore, we opted for a larger sampling grid, with 100 cells on the time axis and 80 cells on the frequency axis, which yielded the acquisition of 8000 semilandmarks from the sound surface, a 90% reduction in data dimensionality (Figure 28 and 29).

R code: Once the background amplitude is reassigned for the chosen value of -25 dB, we then create new sequences to use as coordinates in the new sampling grid. Next, the new sequences are used to subset the matrix of amplitude values, reducing its dimensionality and spectrogram resolution. The new coordinates are plotted for visualization and the subsetting matrices are used to produce 3D spectrograms with reduced dimensionality and resolution (Figure 29).

```
# Reassigning background amplitude values
for(i in 1:length(s.eph$amp)){if(s.eph$amp[i] == -Inf |
s.eph$amp[i] <= -25)
{s.eph$amp[i] <- -25}} # P. centralis
for(i in 1:length(s.kro$amp)){if(s.kro$amp[i] == -Inf |
s.kro$amp[i] <= -25)
{s.kro$amp[i] <- -25}} # P. kroyeri

# Create new sequence to use as coordinates. Any spectrogram
object can be used at this point
time.seq <- seq(1, length(s.eph$time), length = 100)
freq.seq <- seq(1, length(s.eph$freq), length = 80)

# Subset original coordinates by new sequence
time.sub <- s.eph$time[time.seq]
freq.sub <- s.eph$freq[freq.seq]

# Subset each matrix of amplitude values using new sequences
amp.eph.sub <- s.eph$amp[freq.seq, time.seq] # P. centralis
amp.kro.sub <- s.kro$amp[freq.seq, time.seq] # P. kroyeri

# Visualize sampling grid of semilandmark coordinates
plot(x=c(0,4), y=c(0,6), type='n', las=1, xlab="Time (s)",
ylab="Frequency (kHz)",
```

```
main="Simplified grid of semilandmark coordinates")
abline(v=time.sub, h=freq.sub, lwd=0.1)
```

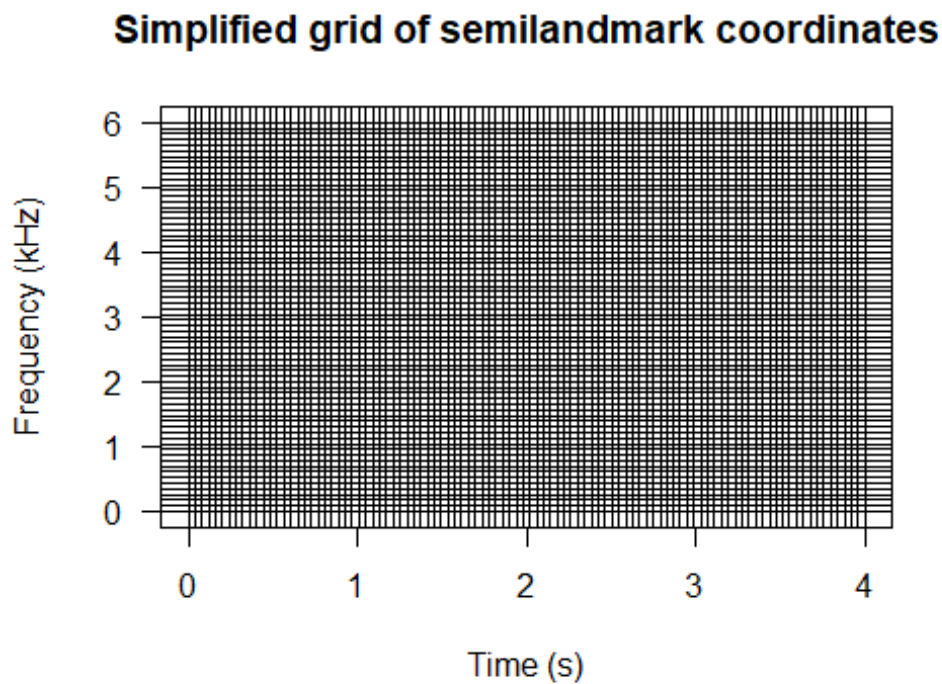


Figure 28: Simplified grid of semilandmark coordinates with 100 cells on the time X-axis and 80 cells on the frequency Y-axis, enabling the acquisition of 8000 semilandmarks from the sound surface.

```
# Simplified 3D spectrogram using new sampling grid
par(mfrow=c(1,2))
persp3D(x=time.sub, y=freq.sub, z=t(amp.eph.sub), theta=30,
phi=35, resfac=1, r=3, expand=0.5, scale=T, axes=T,
ticktype="detailed", nticks=4, cex.axis=0.7, border="black",
lwd=0.1, col=spectro.colors(n=100), xlab="Time (s)",
ylab="Frequency (kHz)", zlab="Amplitude (dB)",
clab=expression('Amplitude dB'),
main=expression(italic('Physalaemus centralis'))))
persp3D(x=time.sub, y=freq.sub, z=t(amp.kro.sub), theta=30,
phi=35, resfac=1, r=3, expand=0.5, scale=T, axes=T,
ticktype="detailed", nticks=4, cex.axis=0.7, border="black",
lwd=0.1, col=spectro.colors(n=100), xlab="Time (s)",
ylab="Frequency (kHz)", zlab="Amplitude (dB)", clab=expression('
Amplitude dB'), main=expression(italic('Physalaemus kroyeri')))
```

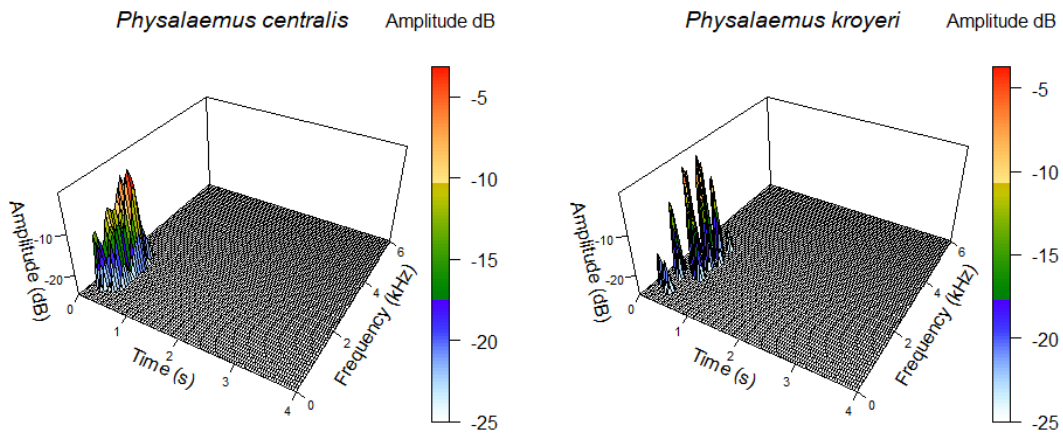


Figure 29: Three dimensional spectrograms resulting from the simplified sampling grid applied to the advertisement calls of *Physalaemus centralis* (left) and *P. kroyeri* (right).

However, this sampling grid resulted in biased results. The majority of *Physalaemus* calls consists in short notes (see Figure 19), which in turn result in a very large “blank” space in the spectrogram output. This blank space can be thought as a series of semilandmarks whose coordinates will always be identical in the majority of analyzed species, whereas a few (*e.g.* *P. olfersii*, *P. lisei*, *P. soaresi*) will be nicely sampled throughout the sound window. In our case study, the linear sequences employed as sampling grid prioritized the few species with long duration calls, whereas the remaining species had remarkably similar semilandmark configurations.

As a way around this issue, we employed a logarithmic (`log`) sequence on the time axis (Figure 30 and 31) instead of the linear grid employed by MacLeod et al. (2013). This emphasized the short duration calls whilst also encompassing the long ones.

R code: Similarly to the lask code chunk, we first create a log sequence to use and then use it to subset the original time coordinates and the matrix of amplitude values. The new coordinates are plotted for visualization (Figure 30) as well as 3D spectrograms resulting from the new matrices (Figure 31).

```
# Create log sequence
time.lseq <- rpql::lseq(1, length(s.eph$time), length = 100)

# Subset coordinates and matrices using log sequence
time.lsub <- s.eph$time[time.lseq]

amp.eph.lsub <- s.eph$amp[freq.seq, time.lseq] # P. centralis
```

```

amp.kro.lsub <- s.kro$amp[freq.seq, time.lseq] # P. kroyeri

# Visualize sampling grid using logarithmic sequence
plot(x=c(0,4), y=c(0,6), type='n', las=1, xlab="Time (s)",
ylab="Frequency (kHz)", main="Sampling grid using log scale")
abline(v=time.lsub, h=freq.sub, lwd=0.1)

```

Sampling grid using log scale

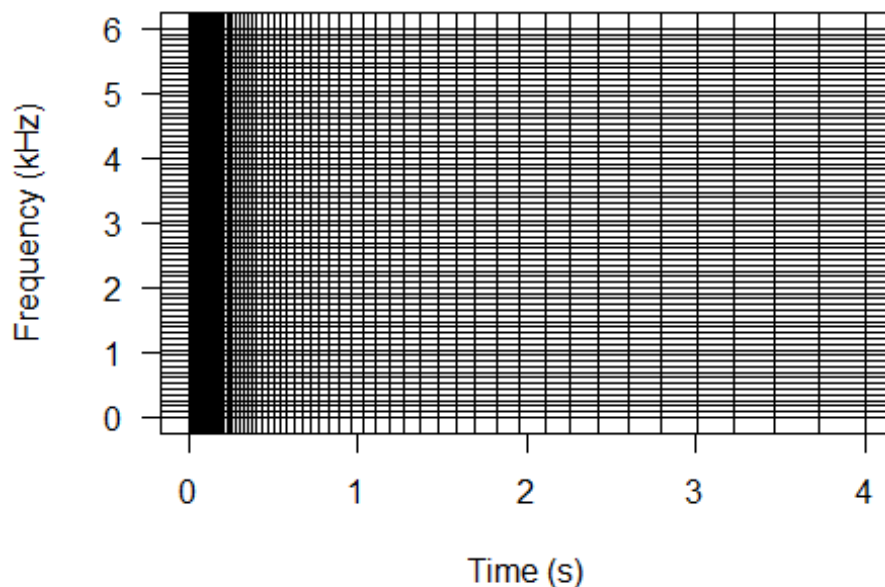


Figure 30: Simplified grid of semilandmark coordinates using a logarithmic sequence on the time X-axis. Grid encompass 100 cells on the time X-axis and 80 cells on the frequency Y-axis, enabling the acquisition of 8000 semilandmarks from the sound surface.

```

# Simplified 3D spectrogram using log scale
par(mfrow=c(1,2))
persp3D(x=time.lsub, y=freq.sub, z=t(amp.eph.lsub), theta=30,
phi=35, resfac=1, r=3, expand=0.5, scale=T, axes=T,
ticktype="detailed", nticks=4, cex.axis=0.7, border="black",
lwd=0.1, col=spectro.colors(n=100), xlab="Time (s)",
ylab="Frequency (kHz)", zlab="Amplitude (dB)",
clab=expression('Amplitude dB'),
main=expression(italic('Physalaemus centralis'))))
persp3D(x=time.lsub, y=freq.sub, z=t(amp.kro.lsub), theta=30,
phi=35, resfac=1, r=3, expand=0.5, scale=T, axes=T,
ticktype="detailed", nticks=4, cex.axis=0.7, border="black",
lwd=0.1, col=spectro.colors(n=100), xlab="Time (s)",

```

```
ylab="Frequency (kHz)", zlab="Amplitude (dB)",
clab=expression('Amplitude dB'),
main=expression(italic('Physalaemus kroyeri')))
```

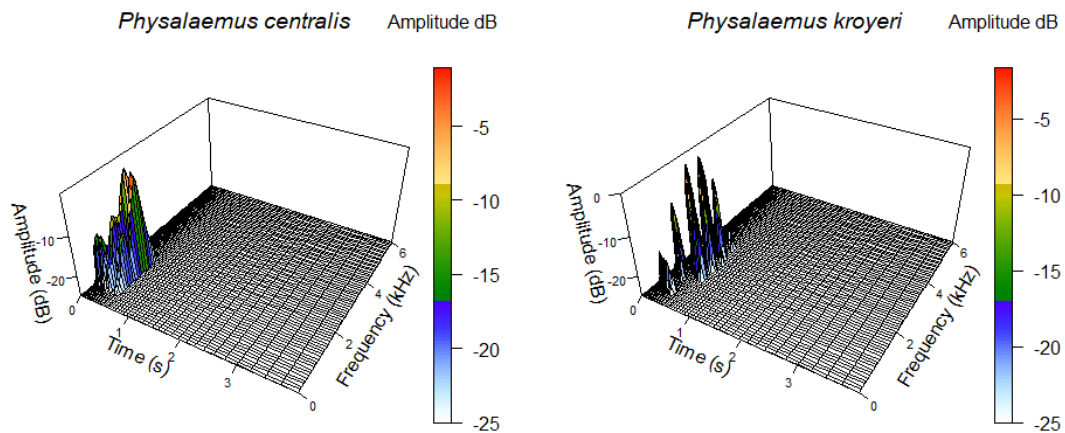


Figure 31: Three dimensional spectrograms resulting from the simplified sampling grid using a logarithmic sequence on the time X-axis, and applied to the advertisement calls of *Physalaemus centralis* (left) and *P. kroyeri* (right).

Ultimately, we can transform our resulting amplitude matrix (dimensions = 80 x 100) into a new matrix of three dimensional coordinates (dimensions = 8000 x 3), with each column representing an axis in a 3D plot. The new matrix are used to create a 3D plot of semilandmarks as points (Figure 32 and Figure 33).

R code: In order to transform the amplitude matrix into a matrix of semilandmark coordinates, we first assign the vectors of X and Y coordinates respectively as the column and row names in our subsetting matrix. This is done for both linear and logarithmic matrices. A new matrix is then created using function `melt` (`reshape2` package) and the column names of set using `setNames`. Note that the matrix has to be transposed (*i.e.* set columns as rows, and rows as columns) to prevent errors between X and Y axis. The new matrix of semilandmark coordinates are used to produce three dimensional scatterplots using `scatter3D` function (`plot3D` package).

```
# Assign time and frequency coordinates as column and row
names of amplitude matrix

# Physalaemus centralis
colnames(amp.eph.sub) <- time.sub
rownames(amp.eph.sub) <- freq.sub
```

```

colnames(amp.eph.lsub) <- time.lsub
rownames(amp.eph.lsub) <- freq.sub

# Physalaemus kroyeri
colnames(amp.kro.sub) <- time.sub
rownames(amp.kro.sub) <- freq.sub

colnames(amp.kro.lsub) <- time.lsub
rownames(amp.kro.lsub) <- freq.sub

# Transform amplitude matrix into 3D coordinates

# Physalaemus centralis
eph.3D <- as.matrix(setNames(reshape2::melt(t(amp.eph.sub)),
c('time', 'freq', 'amp'))
eph.3D.log<-as.matrix(setNames(reshape2::melt(t(amp.eph.lsub)),
c('time', 'freq', 'amp'))

# Physalaemus kroyeri
kro.3D <- as.matrix(setNames(reshape2::melt(t(amp.kro.sub)),
c('time', 'freq', 'amp'))
kro.3D.log<-as.matrix(setNames(reshape2::melt(t(amp.kro.lsub)),
c('time', 'freq', 'amp'))

# Three dimensional plot of semilandmarks as points using
linear scale
par(mfrow=c(1,2))
scatter3D(x=eph.3D[,1], y=eph.3D[,2], z=eph.3D[,3], theta=30,
phi=35, resfac=1, r=3, expand=0.5, scale=T, axes=T,
ticktype="detailed", nticks=4, cex.axis=0.7,
col=spectro.colors(n=100), pch=21, cex=0.4, xlab="Time (s)",
ylab="Frequency (kHz)", zlab="Amplitude (dB)",
clab=expression('Amplitude dB'),
main=expression('3D semilandmarks'))
scatter3D(x=kro.3D[,1], y=kro.3D[,2], z=kro.3D[,3], theta=30,
phi=35, resfac=1, r=3, expand=0.5, scale=T, axes=T,
ticktype="detailed", nticks=4, cex.axis=0.7,

```

```
col=spectro.colors(n=100), pch=21, cex=0.4, xlab="Time (s)",
ylab="Frequency (kHz)", zlab="Amplitude (dB)",
clab=expression('Amplitude dB'),
main=expression('3D semilandmarks'))
```

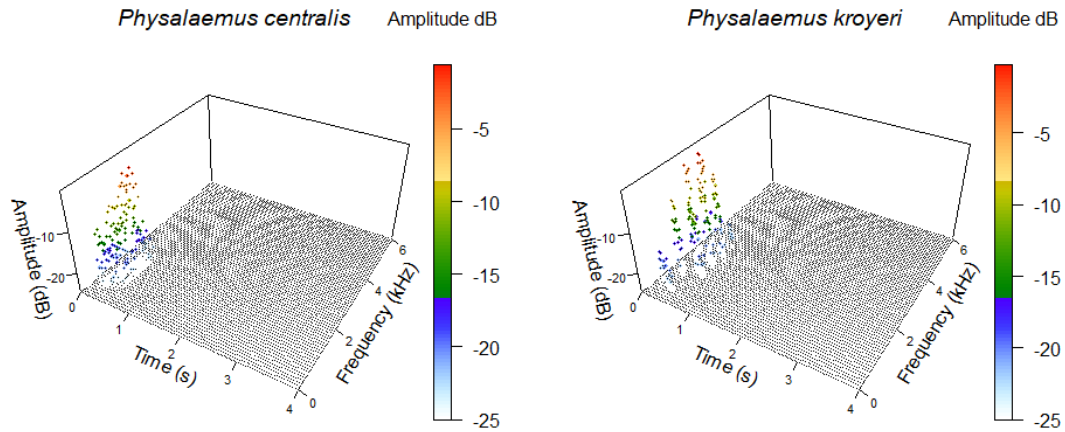


Figure 32: Three dimensional view of spectrograms as points, representing the semilandmarks acquired from a linear sampling grid. Advertisement calls of *Physalaemus centralis* (left) and *P. kroyeri* (right).

```
# Three dimensional plot of semilandmarks as points using
logarithmic scale
par(mfrow=c(1,2))
scatter3D(x=eph.3D.log[,1],y=eph.3D.log[,2], z=eph.3D.log[,3],
theta=30, phi=35, resfac=1, r=3, expand=0.5, scale=T, axes=T,
ticktype="detailed", nticks=4, cex.axis=0.7,
col=spectro.colors(n=100), pch=21, cex=0.4, xlab="Time (s)",
ylab="Frequency (kHz)", zlab="Amplitude (dB)",
clab=expression('Amplitude dB'),
main=expression('3D semilandmarks'))

# Physalaemus kroyeri (log scale)
scatter3D(x=kro.3D.log[,1],y=kro.3D.log[,2], z=kro.3D.log[,3],
theta=30, phi=35, resfac=1, r=3, expand=0.5, scale=T, axes=T,
ticktype="detailed", nticks=4, cex.axis=0.7,
col=spectro.colors(n=100), pch=21, cex=0.4, xlab="Time (s)",
ylab="Frequency (kHz)", zlab="Amplitude (dB)",
clab=expression('Amplitude dB'),
main=expression('3D semilandmarks'))
```

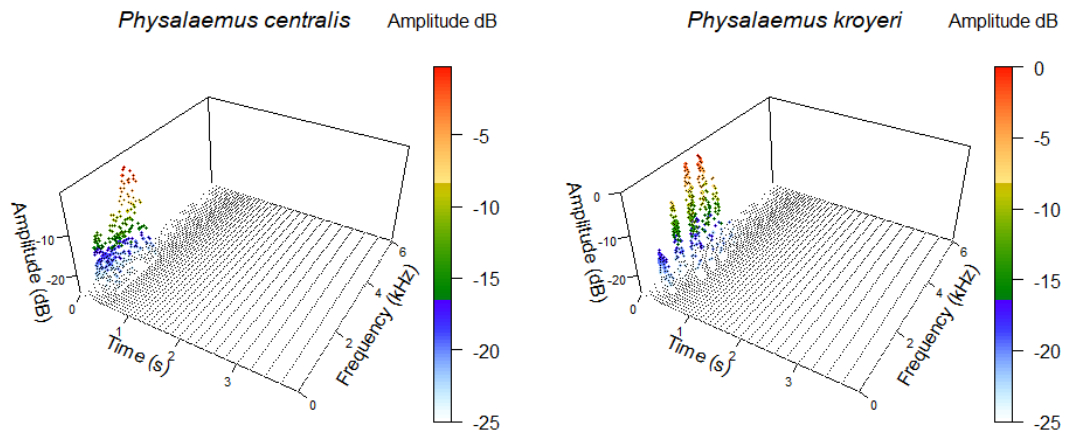


Figure 33: Three dimensional view of spectrograms as points. Each point represents a semilandmark acquired from a logarithmic sampling grid. Advertisement calls of *Physalaemus centralis* (left) and *P. kroyeri* (right).

Eigensound function: Acquiring 3D semilandmarks from multiple wave files

To facilitate the acquisition of semilandmarks from the sound surface of multiple wave files, we introduce new function `eigensound`, which work similarly as `SoundShape`. As default, the function will compute 8000 semilandmarks using a logarithmic scale on the time X-axis, thus prioritizing sounds with short duration rather than long ones.

As with `SoundShape`, the result of `eigensound` can be assigned to an R object or exported as a `tps` file, which can be used by numerous softwares of geometric analysis of shape, such as the TPS series (Rohlf, 2015). The new function also enable its user to export three dimensional plots of semilandmarks either a simplified sound surface or as points. Plots can be exported as `jpeg` (compressed image) or `tiff` (uncompressed image) file format.

R code: In order to use `eigensound`, one must first import it into the R environment. As described for the `frequencies`, `ZC` and `SoundShape` functions, this can be done by pasting the whole code below into R's console and then executing it, or by sourcing the function into R (*i.e.* `source("insert.filepath.here/eigensound.R")`).

Table 3: Arguments to be used by `eigensound` function

Arguments	Description
<code>f</code>	Sampling frequency of wave files (in Hz). By default = 44100.
<code>wl</code>	Length of the window for the analysis. By default = 512.
<code>ovlp</code>	Overlap between two successive windows (in %) for increased spectrogram resolution. By default = 70.
<code>dBlevel</code>	Absolute amplitude value to be used as relative amplitude contour, which will serve as reference for semilandmark acquisition.
<code>flim</code>	Modifications of the frequency Y-axis limits. Vector with two values in kHz.
<code>tlim</code>	Modifications of the time X-axis limits. Vector with two values in seconds.
<code>x.length</code>	Length of sequence to be used as sampling grid coordinates on the time X-axis. <i>i.e.</i> number of cells per side on sound window. By default= <code>x.length</code> =100.
<code>y.length</code>	Length of sequence to be used as sampling grid coordinates on the time Y-axis. <i>i.e.</i> number of cells per side on sound window. By default= <code>y.length</code> =80.
<code>log.scale</code>	A logical, if <code>TRUE</code> , <code>eigensound</code> will use a log scale on the time X-axis, which emphasize short duration sounds and should be preferred when the analyzed sounds present great variation on the time X-axis. If <code>FALSE</code> , a linear scale is used instead (same as MacLeod et al. 2013).
<code>wav.at</code>	Filepath to the folder where wave files are stored. Should be presented between quotation marks. By default, <code>eigensound</code> will use current working directory.
<code>store.at</code>	Filepath to the folder where <code>tps</code> file and spectrogram plots will be stored. Should be presented between quotation marks. By default, <code>eigensound</code> will use current working directory.
<code>plot.exp</code>	A logical, if <code>TRUE</code> , <code>eigensound</code> will export a file for each wave file in the indicated folder. Plots consist in three dimensional images of semilandmarks represented either as points or as sound surface, which is specified by <code>plot.as</code> .
<code>plot.as</code>	Desired format for image files when <code>plot.exp</code> = <code>TRUE</code> . "jpeg" will generate compressed images for quick inspection of semilandmarks, whereas "tiff" or "tif" will generate uncompressed high resolution images that can be edited and used for publication.
<code>plot.type</code>	Desired three dimensional spectrogram when <code>plot.exp</code> = <code>TRUE</code> . "points" will reproduce semilandmarks as points in a three dimensional graph, whereas "surface" will create a simplified sound surface from the calculated semilandmarks (same output employed by MacLeod et al. 2013)
<code>TPS.file</code>	Desired name for <code>tps</code> file containing semilandmark coordinates. Should be presented between quotation marks. By default = <code>NULL</code> , which prevents <code>eigensound</code> from creating a <code>tps</code> file.

```

#~~~~~#
#           eigensound function:           #
#   computing semilandmarks from 3D spectrograms surface   #
#~~~~~#

eigensound <- function(f=44100, wl=512, ovlp=70, dBlevel=25,
flim=c(0,6), tlim=c(0,4), x.length=100, y.length=80,
wav.at=getwd(), store.at=getwd(), plot.exp=TRUE, plot.as="jpeg",
plot.type = "points",TPS.file = NULL, log.scale=TRUE){

  # Require packages
  if(!library("reshape2", logical.return=T)){install.packages
("reshape2")}
  if(!library("seewave", logical.return=T)){install.packages
("seewave")}
  if(!library("tuneR", logical.return=T)){install.packages
("tuneR")}
  if(!library("abind", logical.return=T)){install.packages
("abind")}
  if(!library("plot3D", logical.return=T)){install.packages
("plot3D")}

  require("reshape2")
  require("seewave")
  require("tuneR")
  require("abind")
  require("plot3D")

  # Create TPS file before analysis
  if(!is.null(TPS.file)){
    TPS.path <- paste(store.at, "/", TPS.file, ".tps", sep="")
    file.create(TPS.path, showWarnings = TRUE)
  }

  # Loop for acquiring semilandmark coordinates for each wave
  file on a given folder
  for(file in list.files(wav.at, pattern = ".wav"))
  {

```

```

eig <- readWave(file)

# Store spectrogram as R object
e <- spectro(eig, f=f, wl=wl, ovlp=ovlp, flim=flim,
tlim=tlim, plot=F)

# Create sequences to use as new grid
freq.seq <- seq(1, length(e$freq), length = y.length)

ifelse(isTRUE(log.scale),
      time.seq <- 10^(seq(log10(1), log10(length(e$time)),
length.out = x.length)), #log
      time.seq<-seq(1,length(e$time),length.out=x.length))
# linear scale

# Subset original coordinates by new sequences
time.sub <- e$time[time.seq]
freq.sub <- e$freq[freq.seq]

# Subset matrix of amplitude values using new sequences
amp.sub <- e$amp[freq.seq, time.seq]

# Reassign background amplitude values
for(i in 1:length(amp.sub)){if(amp.sub[i]== -Inf|amp.sub[i]
<= -dBlevel)
{amp.sub[i] <- -dBlevel}}

# Assign time and frequency coordinates as column and row
names of amplitude matrix
colnames(amp.sub) <- time.sub
rownames(amp.sub) <- freq.sub

# Transform amplitude matrix into semilandmark 3D coordinates
ind.3D <- as.matrix(setNames(reshape2::melt(t(amp.sub)),
c('time', 'freq', 'amp'))))

# Store semilandmark coordinates in array
ifelse(!exists('coord3D'),

```

```

coord3D <- array(data=ind.3D, dim=c(dim(ind.3D), 1)),
coord3D <- abind::abind(coord3D, ind.3D, along=3)

# Plot semilandmarks as points or sound surface
if(plot.exp==TRUE){
  if(plot.as == "jpeg"){jpeg(width =5000,height = 3500,
units = "px", res = 500,
  filename=paste(store.at,"/", sub(".wav","",file),
".jpg", sep=""))} # compressed images
  if(plot.as=="tiff"|plot.as=="tif"){tiff(width=5000,
height=3500, units="px", res=500, filename=paste(store.at, "/",
sub(".wav", "", file), ".tif", sep=""))} # uncompressed image
s
  if(plot.type=="surface"){persp3D(x=time.sub, y=freq.sub,
z=t(amp.sub), border="black", lwd=0.1, theta=30, phi=35,
resfac=1, r=3,expand=0.5, cex.axis=0.7, scale=T, axes=T,
col=spectro.colors(n=100), ticktype="detailed", nticks=4,
xlab="Time (s)", ylab="Frequency (kHz)", zlab="Amplitude (dB)",
main= sub(".wav", "", file), clab=expression('Amplitude dB'))}
  if(plot.type=="points"){scatter3D(x=ind.3D[,1],
y=ind.3D[,2], z=ind.3D[,3], pch=21, cex=0.5, theta=30, phi=35, r
esfac=1, r=3, expand=0.5, cex.axis=0.7, scale=T, axes=T,
col=spectro.colors(n=100), ticktype="detailed", nticks=4,
xlab="Time (s)", ylab="Frequency (kHz)", zlab="Amplitude (dB)",
main= sub(".wav", "", file), clab=expression('Amplitude dB'))}

  dev.off()
} #end plot

# Export semilandmark coordinates as TPS file
if(!is.null(TPS.file)){
  lmline <- paste("LM=", dim(ind.3D)[1], sep="")
  write(lmline, TPS.path, append = TRUE)
  write.table(ind.3D, TPS.path, col.names = FALSE,
row.names = FALSE, append =TRUE)
  idline <- paste("ID=", sub(".wav", "", file), sep="")
  write(idline, TPS.path, append = TRUE)

```

```

    write("", TPS.path, append = TRUE)
    rm(lmline, idline)
  } #end tps file
} #end loop

# Assign names to arrays' third dimension
dimnames(coord3D)[[3]] <- sub(".wav", "", list.files(wav.at,
pattern = ".wav"))
results <- coord3D

} #end function

```

Using eigensound function

In order to illustrate how to use `eigensound` function, we selected a representative note of each *Physalaemus* species and stored them in a single folder. For illustrative purposes, we define this folder as our working directory, but store the spectrogram plots and the tps file elsewhere by specifying a different path for the outputs (same as with `SoundShape` function).

R code: The working directory is set using the `setwd` function. We call `dir(pattern=".wav")` to visualize wave files stored on the selected folder. Two alternative ways to use `eigensound` function are employed. We first use the function controlling all of its arguments, which are the same as in the default of `eigensound`. Alternatively, we can focus on the arguments we wish to control and `eigensound` will run its default settings. The second alternative is much simpler and we recommend it to avoid typing errors.

```

# Source eigensound function into R's environment
source("insert.filepath.here/eigensound.R")

# Input eigensound function into R environment
source("C:/R/Funcoes/eigensound.R")

# Define working directory where wave files are stored
setwd("C:/R/eigensound-Sample")
dir(pattern=".wav")

```

```
## [1] "P-aguirrei_Linhares-ES_FNJV-0031776.wav"
## [2] "P-albifrons_Maracas-BA_FNJV-0031943.wav"
## [3] "P-albonotatus_Caceres-MT_FNJV-0031171.wav"
## [4] "P-biligonigerus_Osorio-RS_FNJV-0031682.wav"
## [5] "P-bokermanni_Santo-Andre-SP_FNJV-0031184.wav"
## [6] "P-camacan_Igrapiuna-BA_FNJV-0034021.wav"
## [7] "P-cicada_Patos-PB_FNJV-0031191.wav"
## [8] "P-crombiei_Aracruz-ES_FNJV-0032255.wav"
## [9] "P-cuvieri_Sao-Jose-Campos-SP_CBUFG-196.wav"
## [10] "P-deimaticus_Serra-do-Cipo-MG_CBUFG-173.wav"
## [11] "P-centralis_Barra-do-Bugres-MT_FNJV-0031235.wav"
## [12] "P-erythros_Ouro-Preto-MG_CBUFG-152.wav"
## [13] "P-evangelistai_Santana-do-Riacho-MG_FNJV-0031636.wav"
## [14] "P-feioi_Vicosa-MG_FNJV-0031559.wav"
## [15] "P-fischeri_Mantecal_Venezuela_FNJV-0031232.wav"
## [16] "P-gracilis_Porto-Alegre-RS_FNJV-0031609.wav"
## [17] "P-kroyeri_Ilheus-BA_FNJV-0032047.wav"
## [18] "P-lisei_Sao-Francisco-Paula-RS_FNJV-0031645.wav"
## [19] "P-maculiventris_Salesopolis-SP_FNJV-0033756.wav"
## [20] "P-marmoratus_Luis-Antonio-SP_FNJV-0004592.wav"
## [21] "P-moreirae_Salesopolis-SP_FNJV-0033755.wav"
## [22] "P-nanus_Rancho-Queimado-SC_FNJV-0031597.wav"
## [23] "P-nattereri_Caceres-MT_FNJV-0031253.wav"
## [24] "P-obtectus_Linhares-ES_FNJV-0031774.wav"
## [25] "P-olfersii_Sao-Jose-do-Barreiro-SP_FNJV-0031263.wav"
## [26] "P-riograndensis_Eldorado-do-Sul-RS_FNJV-0031643.wav"
## [27] "P-rupestris_Serra-do-Ibitipoca-MG_CBUFG-875.wav"
## [28] "P-signifer_Itaguaí-RJ_FNJV-0031273.wav"
## [29] "P-soaresi_Itaguaí-RJ_FNJV_0031272.wav"
## [30] "P-spiniger_Caraguatatuba-SP_FNJV-0031570.wav"

# Compute semilandmarks for each wave file at working
directory and store them elsewhere
eig.sample <- eigensound(f=44100, wl=512, ovlp=70, dBlevel=25,
tlim = c(0, 4), flim=c(0, 6), x.length=100, y.length=80,
wav.at = getwd(), store.at = "C:/R/eigensound-Sample/Output",
plot.exp = TRUE, plot.as = "jpeg", plot.type = "points",
TPS.file = "eig_sample", log.scale=TRUE)
```

```
# Simpler way to call eigensound function. Arguments omitted  
imply that default will be used  
eig.sample <- eigensound(TPS.file = "eig_sample",  
store.at = "C:/R/eigensound-Sample/Output")
```

R code: We can explore the contents of the eigensound object by calling a few base functions, such as the class of the object, its dimensions dim (the result of dim is in the form (rows, columns, elements)), and the class of each element in the array. Ultimately, we can verify the semilandmark coordinates by calling the eigensound object created.

```
# Verify contents of object created  
class(eig.sample)  
  
## [1] "array"  
  
dim(eig.sample)  
  
## [1] 8000      3      30  
  
class(eig.sample[, , 1])  
  
## [1] "matrix"  
  
# Visualize coordinates of semilandmarks 1000 to 1015 from  
first two elements  
eig.sample[1000:1015, , 1:2]  
  
## , , P-aguirrei_Linhares-ES_FNJV-0031776  
##  
##           time      freq amp  
## [1,] 3.99650655 0.6086957 -25  
## [2,] 0.00000000 0.6956522 -25  
## [3,] 0.00000000 0.6956522 -25  
## [4,] 0.00000000 0.6956522 -25  
## [5,] 0.00000000 0.6956522 -25  
## [6,] 0.00000000 0.6956522 -25  
## [7,] 0.00000000 0.6956522 -25  
## [8,] 0.00000000 0.6956522 -25  
## [9,] 0.00000000 0.6956522 -25  
## [10,] 0.00000000 0.6956522 -25  
## [11,] 0.00000000 0.6956522 -25
```

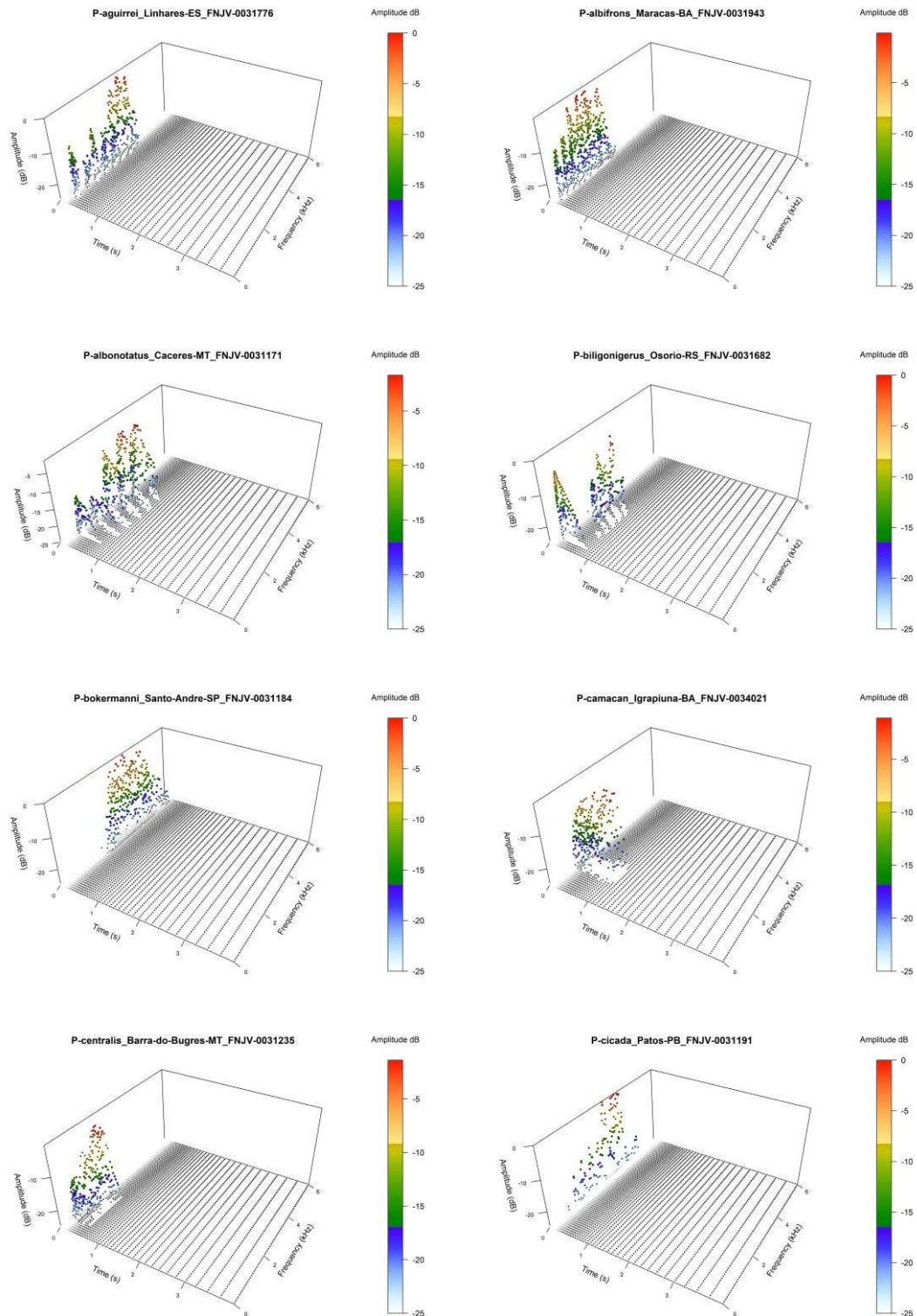
```
## [12,] 0.00349345 0.6956522 -25
## [13,] 0.00349345 0.6956522 -25
## [14,] 0.00349345 0.6956522 -25
## [15,] 0.00349345 0.6956522 -25
## [16,] 0.00349345 0.6956522 -25
##
## , , P-albifrons_Maracas-BA_FNJV-0031943
##
##           time      freq amp
## [1,] 3.99650655 0.6086957 -25
## [2,] 0.00000000 0.6956522 -25
## [3,] 0.00000000 0.6956522 -25
## [4,] 0.00000000 0.6956522 -25
## [5,] 0.00000000 0.6956522 -25
## [6,] 0.00000000 0.6956522 -25
## [7,] 0.00000000 0.6956522 -25
## [8,] 0.00000000 0.6956522 -25
## [9,] 0.00000000 0.6956522 -25
## [10,] 0.00000000 0.6956522 -25
## [11,] 0.00000000 0.6956522 -25
## [12,] 0.00349345 0.6956522 -25
## [13,] 0.00349345 0.6956522 -25
## [14,] 0.00349345 0.6956522 -25
## [15,] 0.00349345 0.6956522 -25
## [16,] 0.00349345 0.6956522 -25
```

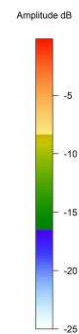
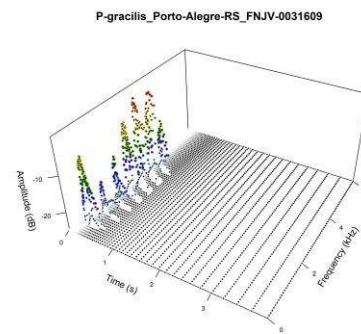
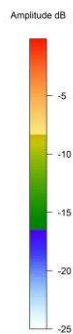
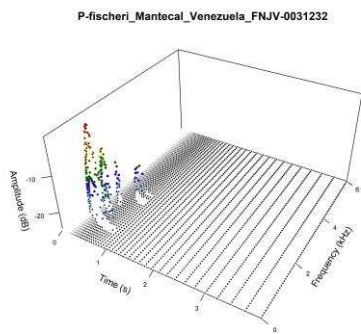
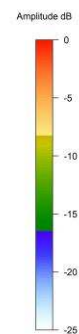
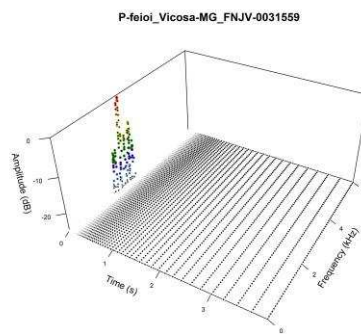
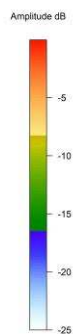
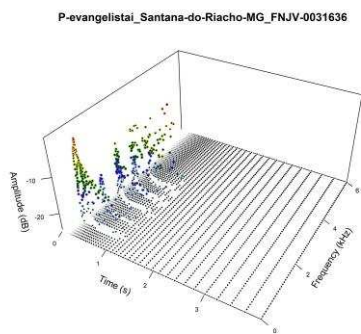
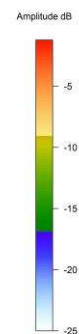
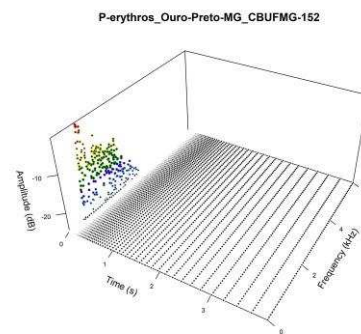
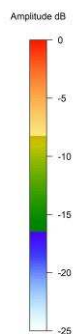
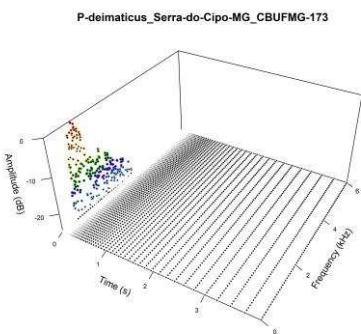
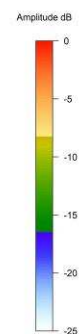
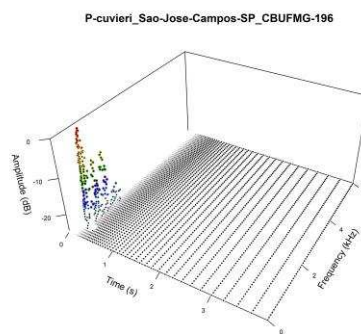
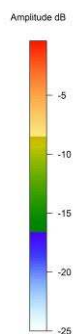
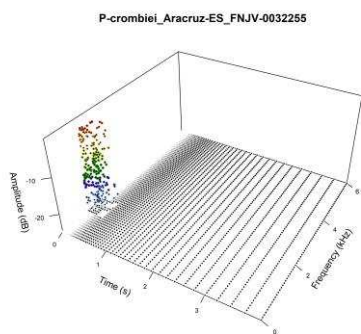
By assinging the result of `eigensound` to an R object, semilandmark coordinates are stored as a 3D array, which is a special type of `list` that can be though as a filing cabinet, in which the `array` is the cabinet and each element is an archive. This special list is will be used by by `geomorph` package in the subsequent steps of the analysis. In our sample, `eig.sample` is an array composed of 30 elements (*i.e.* 30 species of *Physalaemus*), each one is a `matrix` with 8000 rows and 3 columns (*i.e.* x, y and z coordinates of 8000 semilandmarks).

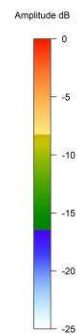
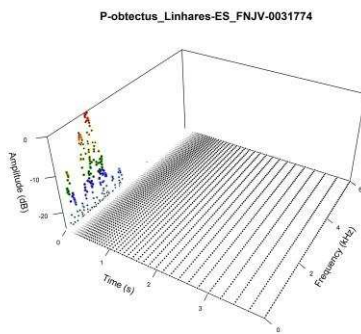
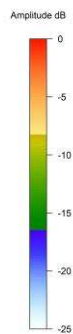
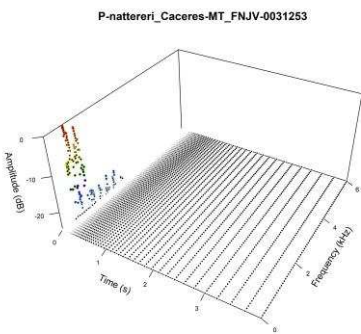
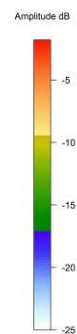
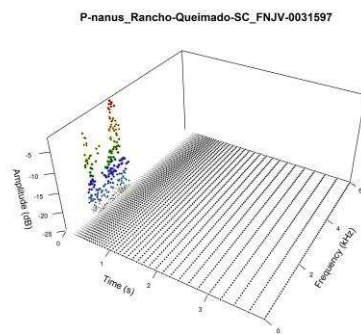
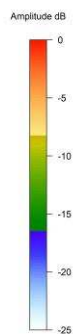
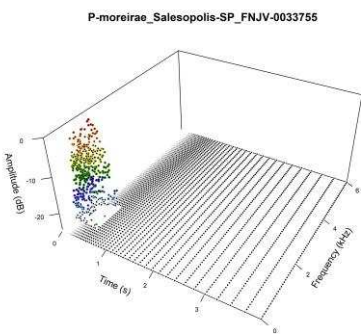
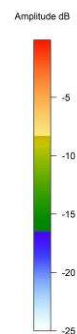
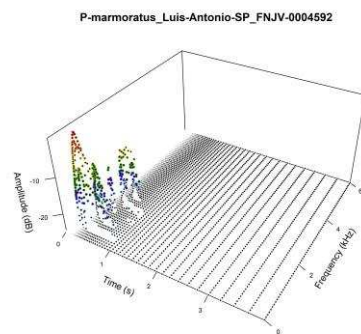
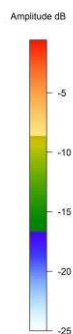
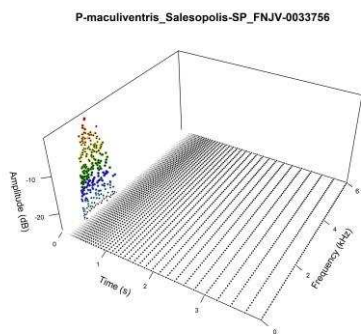
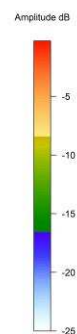
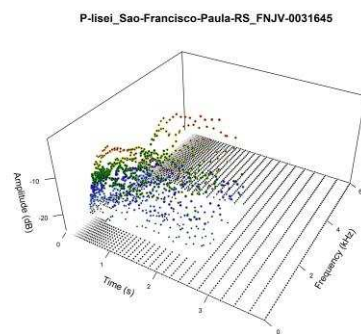
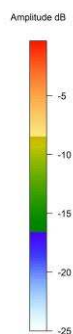
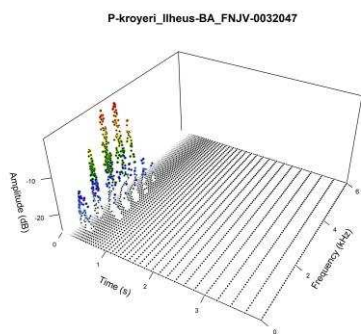
Ultimately, we can now check how semilandmarks were plotted in each species (Figure 34).

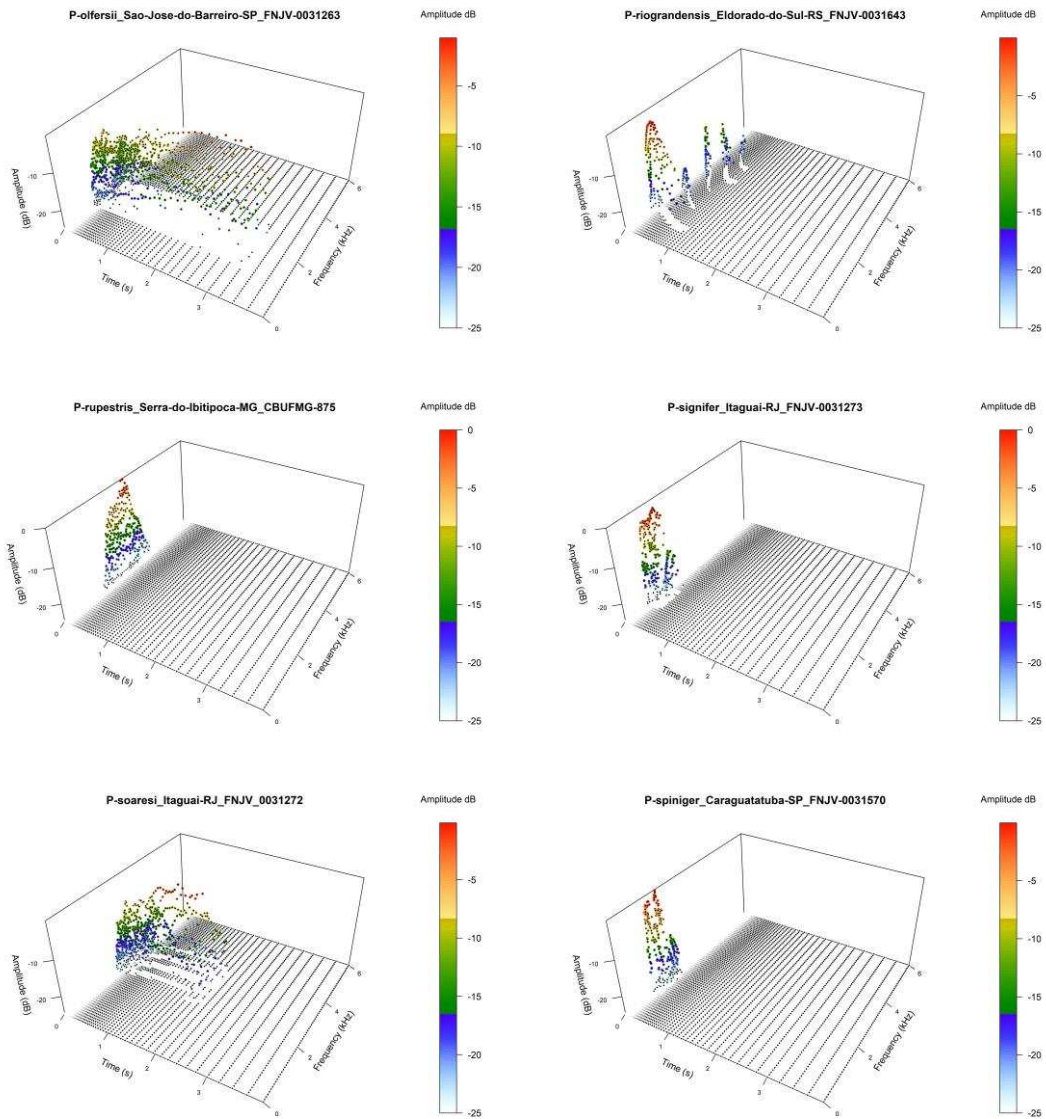
Figure 34: (Displayed below) Three dimensional view of spectrograms as points. Each point represents a semilandmark acquired from a logarithmic sampling grid and each plot is an stereotyped advertisement call of 30 *Physalaemus* species. These species had a representative call/note selected previously and spectrograms were generated automatically by the eigensound function.

```
# Display spectrograms from folder
include_graphics(list.files("C:/R/eigensound-Sample/Output",
pattern=".jpg", full.names = TRUE))
```









Now that we managed to acquire semilandmarks from the shape of two dimensional spectrograms and also from the three dimensional surface of it, we can leave the spectrograms aside and proceed with the analysis regarding only the semilandmarks. The final segment of this supplementary file will focus on the Geometric Morphometric analysis of sound shape, which represents a tool for taxonomic and evolutionary studies of bioacoustics.

Geometric Morphometrics applied to Bioacoustics

Geometric Morphometrics (GM) can be summarized as the comparative analysis of shape, in which the concept of *shape* represent the base for the mathematical theory behind it (Zelditch et al., 2012). By definition, shape is all the geometric information from a configuration of points (i.e. landmarks or semilandmarks) that are invariant to changes in translation, rotation, and scale (Bookstein, 1991). The shape is generally represented by a set of landmarks (LM), which are often referred as “homologous points” and defined as points of correspondence that matches between and within populations for a given object (Dryden & Mardia, 1998; Zelditch et al., 2012). The matching idea is not strictly biological though, but also a mathematical constraint. If the landmarks do not correspond, we cannot compare shapes (Zelditch et al., 2012).

Both `eigensound` and `SoundShape` functions are meant for the acquisition of semilandmarks (SM) directly from `wave` files (Figure 35 and 36). Although the methods embedded in these functions differ, both of them provide its user with the option of storing the results as an R object, enabling GM analysis to be performed within R platform. Alternatively, it is also possible to export semilandmark coordinates as a `tps` file, which can be used by any of the numerous softwares of geometric analysis of shape, such as the TPS series (Rohlf, 2015). Herein, we will conduct the analysis within R’s environment, thus providing a guidance for future research to be performed strictly on open code.

Defining the Geometric Morphometric methods for bioacoustic shape analysis in R

Geometric Morphometric (GM) analysis is performed with `geomorph` package of R platform, however our study sometimes performed the same analysis using codes similar to or embedded in functions from this package. Those cases will be explicitly stated throughout this section and the recommended functions from `geomorph` will be provided. For a complete guide to these packages, please see Adams et al. (2017) and Sherratt (2017).

During the early phases of sound shape analysis, we will explore our data using only a representative advertisement call from of each of the 30 *Physalaemus* species (Figures 19 and 34; more details in Rocha & Romano, *in prep*). Important to note that the

call of those species comprise a wide variety of sound shapes, which ended up with some peculiar results. Although these results are briefly discussed for illustrative purposes, the sample employed in this exploratory phase should not be taken on a high note. However, the results provide useful information about the application of both methods and visualization of the results.

Note: As described in previous sections, both *SoundShape* and *eigensound* analysis are conducted with calls set to begin at time = 0. Therefore, the semilandmark coordinates of *SoundShape* and *eigensound* functions have been recreated using the same wave files.

R code: First, we recreate both *eigensound* and *SoundShape* objects using the same wave files by inputting the same file path. The results are used to calculate the mean shape configuration *mshape* for the complete set of semilandmarks computed in each approach. The very large amount of semilandmarks acquired by *eigensound* prevents the clear visualization of the mean shape along with the complete set of semilandmarks, therefore mean shape and the set of semilandmarks are plotted separately. Semilandmarks from *SoundShape* are plotted along with mean shape using *plot* and *points* function from R graphics, however *geomorph* offer the function *plotAllSpecimens*, which provide a simpler alternative for plotting a complete set of semilandmarks in both two and three dimensions.

```
# Acquire semilandmarks for each wave file on folder
SS.sample <- SoundShape(wav.at = "C:/R/Physalaemus-Sample",
plot.exp = FALSE)

eig.sample <- eigensound(wav.at = "C:/R/Physalaemus-Sample",
plot.exp = FALSE)

# Calculate mean shape for each set of semilandmarks
SS.mean <- mshape(SS.sample)

eig.mean <- mshape(eig.sample)

# Plot complete set of semilandmarks and mean shape from Sound
Shape
plot(SS.sample[, 1, ], SS.sample[, 2, ], pch=21, cex=0.9,
xlim=c(0,4), ylim=c(0,6), xlab="Time coordinates",
ylab="Frequency coordinates",
```

```
main="Semilandmarks from SoundShape")
points(SS.mean, pch=21, bg="orange", cex=1.1)
```

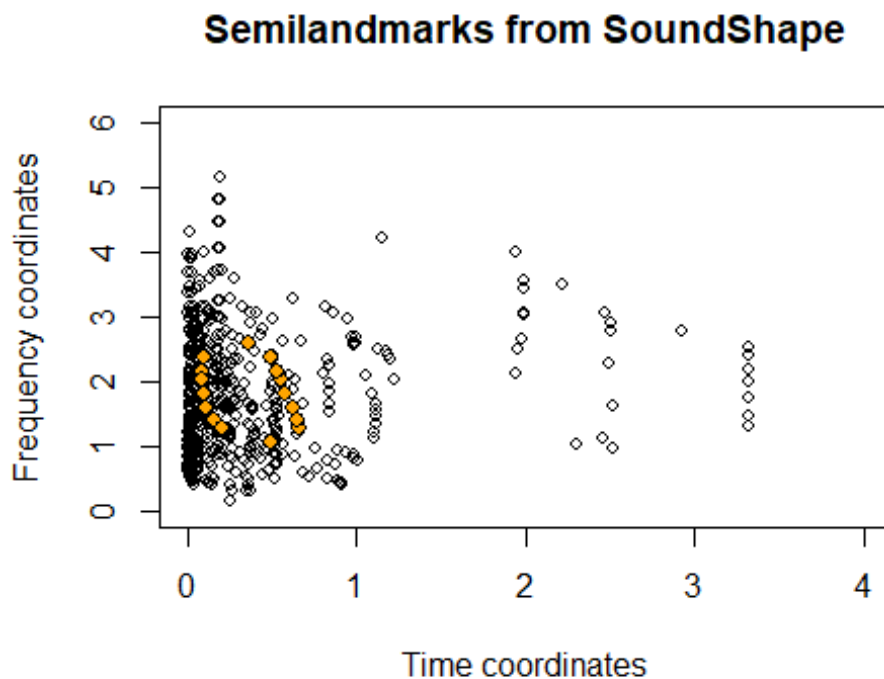


Figure 35: Complete set of semilandmarks (hollow circles) acquired by SoundShape function from representative advertisement calls of 30 *Physalaemus* species. Orange dots represent mean coordinates for each of the 16 semilandmarks calculated from the complete set of semilandmark coordinates.

```
# Plot complete set of semilandmarks and mean shape from
eigen sound
par(mfrow=c(1,2))
scatter3D(x=eig.sample[,1],y=eig.sample[,2],
z=eig.sample[,3], theta=30, phi=35, resfac=1, r=3, expand=0.5,
scale=T, axes=T, cex.axis=0.7, ticktype="detailed", nticks=4,
pch=21, col=spectro.colors(n=100), cex=0.4, xlab="Time (s)",
ylab="Frequency (kHz)", zlab="Amplitude (dB)",
clab=expression('Amplitude dB'),
main=expression('Set of semilandmarks'))

scatter3D(x=eig.mean[,1], y=eig.mean[,2], z=eig.mean[,3],
theta=30, phi=35, resfac=1, r=3, expand=0.5, scale=T, axes=T,
cex.axis=0.7, ticktype="detailed", nticks=4, pch=21,
col=spectro.colors(n=100), cex=0.4, xlab="Time (s)",
ylab="Frequency (kHz)", zlab="Amplitude (dB)",
clab=expression('Amplitude dB'), main=expression('Mean shape'))
```

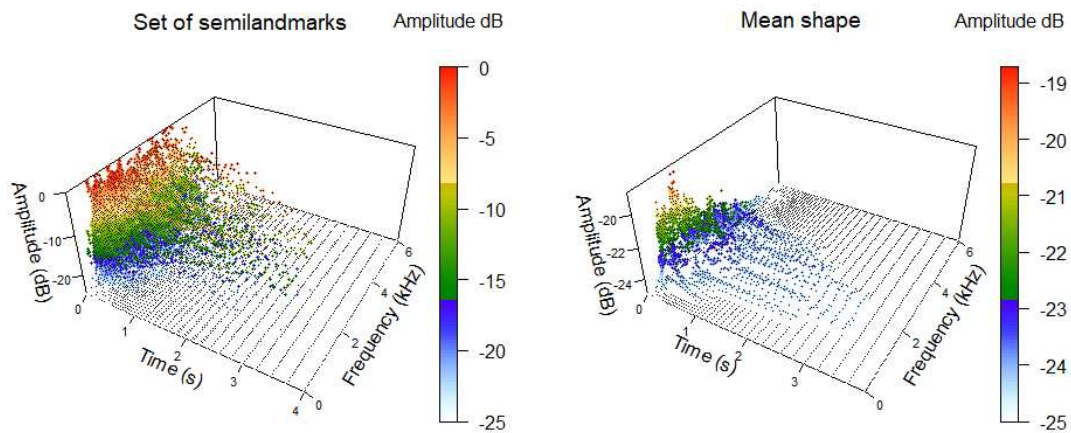


Figure 36: Left: Complete set of three dimensional semilandmarks acquired by `eigensound` function from representative advertisement calls of 30 *Physalaemus* species. Right: Mean coordinates calculated from the set of three dimensional semilandmarks.

Figures 35 and 36 show the complete set of semilandmarks acquired by `SoundShape` and `eigensound`, respectively. Even though these SM configurations represent the shape of sound, they will only be comparable once they had been mathematically translated, rescaled and rotated. One of the main implications of rescaling is that the size in its traditional sense (*e.g.* snout-vent length, call duration) is also removed from the analysis. The geometric scale of landmark configurations is not lost though, and can be calculated using the distances of all landmarks to the center of the form (*i.e.* its “centroid”) in order to achieve the “centroid size”. Even though the centroid size is mathematically independent of shape, it is often biologically correlated with it, which is commonly referred as allometry (Bookstein, 1991).

The Procrustes Superimposition method, in turn, is the series of mathematical procedures aimed to minimize the differences in landmark configuration through translation, rescaling and rotation, without causing any shape alteration (Rohlf & Slice, 1990). The reason to perform these methods is because the set of original semilandmark configurations acquired from spectrograms are not on the same space, even though sounds have been placed at time = 0 s. By performing a Procrustes superimposition method, the differences caused by relative position, orientation and size are removed from the SM configurations. First, each configuration of landmarks is centered at its origin by translation of the centroid. Then, configurations are rescaled by the centroid size (which is fixed at one) and rotated for optimal alignment in relation to the mean shape averaged from the sample (*i.e.* *consensus* shape), thus minimizing the summed squared distances between landmarks (Rohlf & Slice, 1990; Zelditch et al., 2012). Whenever the mean

shape is iteratively calculated from the sample, the whole process is referred as Generalized Procrustes Analysis (GPA). In other words, the 2D and 3D semilandmark configurations acquired from a sound wave already represents the shape of sound. However, these configurations also encompass unwanted information, which can be pragmatically thought as noise and removed from the data by applying a GPA on the original sound shape.

Note: The *eigensound* approach (as originally proposed by MacLeod et al., 2013) does not feature a GPA once the semilandmarks are acquired. According to MacLeod et al. (2013) by computing semilandmarks from the simplified sampling grid, “the spectrogram data are, effectively, shape data that reside as point locations on a high dimensional Kendall shape manifold” (MacLeod et al., 2013). Once these points are acquired, the *eigensound* approach proceed with a principal component analysis followed by a canonical variates analysis. However, in our analysis, we also conducted GPA on the three dimensional spectrogram data in attempt to reduce some of the variance embedded as noise. The original approach is conducted complementarily.

R code: Performing a Generalized Procrustes Analysis in R is quite simple once the semilandmarks have been acquired. The analysis is performed by the function `gpagen`, from `geomorph` package, and we call the argument `PrinAxes=FALSE` to prevent the loss of bioacoustical information through the free rotation of sound shapes. The results are stored as an list, which include the aligned SM coordinates and consensus shape visualized below (Figure 37 and 38).

```
# Perform Generalized Procrustes Analysis on semilandmarks
acquired by SoundShape
SS.gpa<-gpagen(SS.sample,PrinAxes=FALSE, print.progress=FALSE)

# Plot new semilandmark configurations and consensus shape
plot(SS.gpa$coords[,1,], SS.gpa$coords[,2,], pch=21, cex=0.9,
asp=1, xlab="Time coordinates", ylab="Frequency coordinates",
main="Semilandmarks after GPA")
points(SS.gpa$consensus, pch=21, bg="orange", cex=1.1)
```

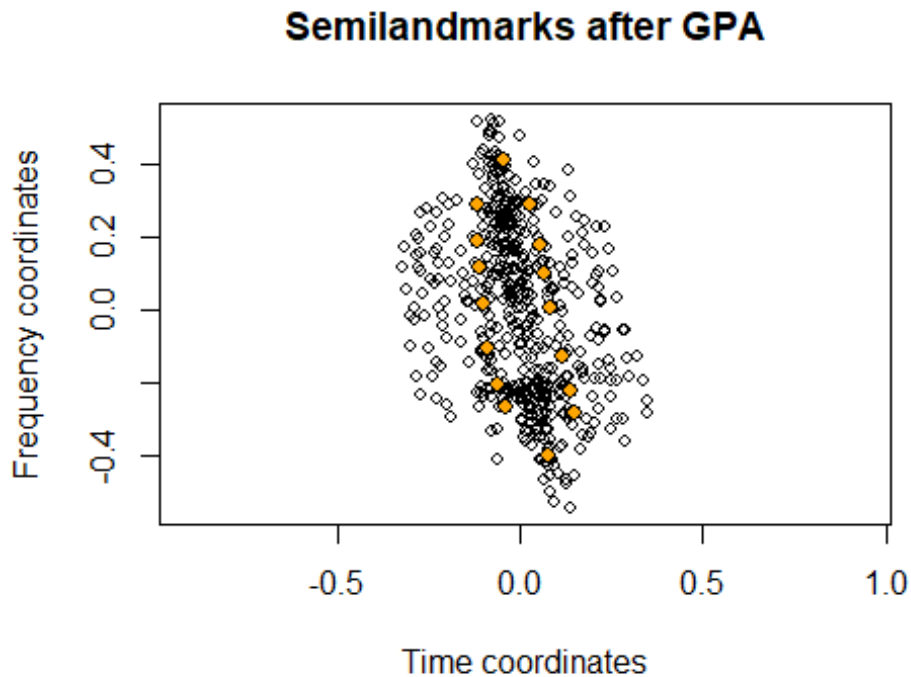


Figure 37: Complete set of semilandmarks (hollow circles) after Generalized Procrustes Analysis. Orange dots represent consensus shape calculated from aligned semilandmarks. Semilandmarks acquired by SoundShape function using representative advertisement calls of 30 *Physalaemus* species.

```
# Perform Generalized Procrustes Analysis on semilandmarks
acquired by eigensound
eig.gpa<-gpagen(eig.sample,PrinAxes=FALSE,print.progress=FALSE)

# Plot new semilandmark configurations and consensus shape
par(mfrow=c(1,2))
scatter3D(x=eig.gpa$coords[,1,], y=eig.gpa$coords[,2,],
z=eig.gpa$coords[, 3,], theta=30, phi=35, resfac=1, r=3,
expand=0.5, scale=T, axes=T, cex.axis=0.7, ticktype="detailed",
nticks=4, pch=21, col=spectro.colors(n=100), cex=0.4,
xlab="Time coordinates",ylab="Frequency coordinates",
zlab="Amplitude coordinates", clab=expression('Amplitude'),
main=expression('Semilandmarks after GPA'))

scatter3D(x=eig.gpa$consensus[,1], y=eig.gpa$consensus[,2],
z=eig.gpa$consensus[,3], theta=30, phi=35, resfac=1, r=3,
expand=0.5, scale=T, axes=T, cex.axis=0.7, ticktype="detailed",
nticks=4, pch=21, col=spectro.colors(n=100), cex=0.4,
xlab="Time coordinates", ylab="Frequency coordinates",
zlab="Amplitude coordinates", clab=expression('Amplitude'),
main=expression('Consensus shape'))
```

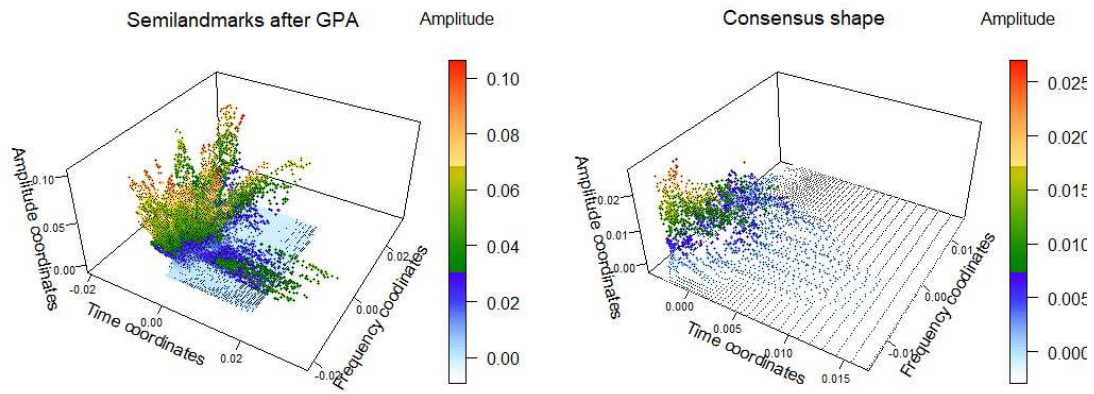


Figure 38: Left: Complete set of three dimensional semilandmarks after Generalized Procrustes Analysis. Right: Consensus shape calculated from the aligned set of three dimensional semilandmarks. Semilandmarks acquired by *eigensound* function using representative advertisement calls of 30 *Physalaemus* species.

Even though the points acquired in both *SoundShape* and *eigensound* approaches are considered as semilandmarks herein and in MacLeod et al. (2013), the GPA performed above deals with these points no differently than it would with landmarks. Particularly for the *SoundShape* approach, the semilandmarks are meant to represent a curve of relative amplitude (see Figures 5 to 18). Therefore, it is important to incorporate information about the curve by defining a few semilandmarks to slide along the curve. In our study, we allowed semilandmarks 2 to 8, and 10 to 16 to slide (see Figures 13 to 16 for semilandmark numbers), whereas semilandmarks 1 and 9 (*i.e.* SM-min and SM-max) were leaved as fixed landmarks. Given the very large amount of semilandmarks from the *eigensound* approach, it is not biologically nor computationally feasible to define sliding semilandmarks in this method.

When conducting a GPA, there are a few different approaches to slide semilandmarks along an estimated tangent at each sliding point. These are aimed for the reduction of noise in the data and result in better alignment (Bookstein, 1997; Sheets et al., 2006; Zelditch et al., 2012; Rohlf, 2015). *geomorph* package feature the two main methods for sliding semilandmarks: minimum Procrustes distance and minimum bending energy. The first one slides the semilandmarks along their tangent directions, minimizing the distance between the adjusted position and the corresponding point in the consensus. The later slide the semilandmarks minimizing the bending energy required for a deformation in the consensus shape (Bookstein, 1997; Zelditch et al., 2012; Rohlf, 2015; Adams et al., 2017).

R code: In order to allow the sliding of semilandmarks along the curve of relative amplitude from `SoundShape`, we create a matrix of semilandmarks numbers using the function `define.sliders`. Then, a new GPA is conducted applying both methods and using the matrix created. Minimum Procrustes distance is set by the argument `ProcD=TRUE`, whilst minimum bending energy is set by `ProcD=FALSE`. The resulting aligned coordinates (Figure 39) are plotted similarly as in the last code chunk.

```
# Define sliders
sliders.SS=rbind(define.sliders(1:9),define.sliders(c(9:16,1)))

# GPA with semilandmarks allowed to slide using Procrustes distance criterion
SS.gpa.Proc<-gpagen(SS.sample,PrinAxes=FALSE,
print.progress = FALSE, curves = sliders.SS, ProcD = TRUE)

# GPA with semilandmarks allowed to slide using minimum bending energy criterion
SS.gpa.Bend <- gpagen(SS.sample, PrinAxes = FALSE,
print.progress = FALSE, curves = sliders.SS, ProcD = FALSE)

# Plot new semilandmark configurations and consensus shape
par(mfrow=c(1,2))
plot(SS.gpa.Proc$coords[,1,], SS.gpa.Proc$coords[,2,], pch=21,
cex=0.9, asp=1, xlab="Time coordinates",
ylab="Frequency coordinates",main="Minimum Procrustes distance")
points(SS.gpa.Proc$consensus, pch=21, bg="orange", cex=1.1)
plot(SS.gpa.Bend$coords[,1,], SS.gpa.Bend$coords[,2,], pch=21,
cex=0.9, asp=1, xlab="Time coordinates",
ylab="Frequency coordinates", main="Minimum bending energy")
points(SS.gpa.Bend$consensus, pch=21, bg="orange", cex=1.1)
```

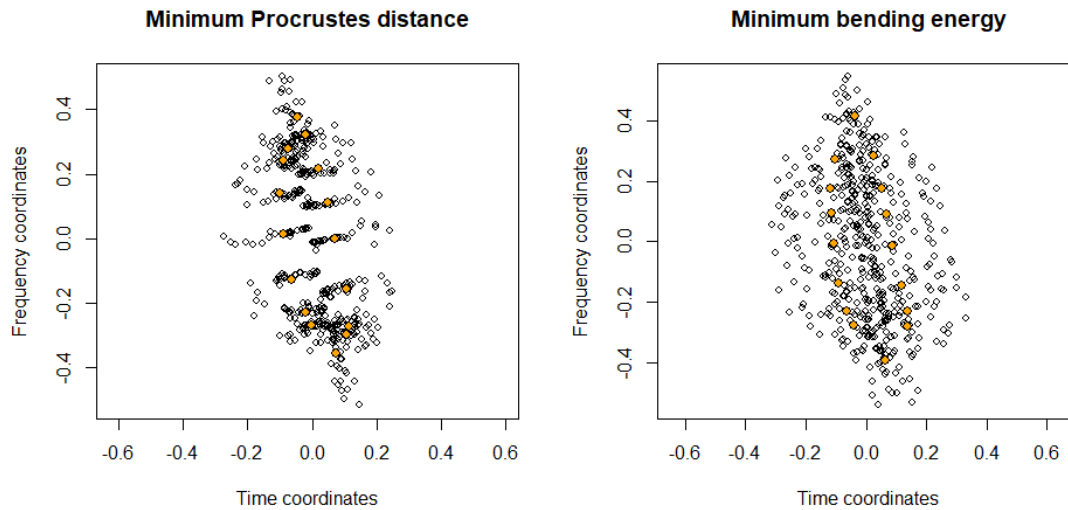


Figure 39: Complete set of semilandmarks (hollow circles) after Generalized Procrustes Analysis using minimum Procrustes distance (left) and minimum bending energy criteria (right). Orange dots represent consensus shape calculated from aligned semilandmarks. Semilandmarks acquired by `SoundShape` function using representative advertisement calls of 30 *Physalaemus* species.

Based on the visual inspection of semilandmark coordinates (Figures 35, 37 and 39), the GPA considering minimum Procrustes distance resulted in the better alignment for the coordinates acquired by `SoundShape` function. Therefore, this method is employed hereafter as we start the ordination methods for the sound shape of 30 *Physalaemus* species.

Principal Components Analysis (PCA) of shape

Once defined the methods from the selection of `wave` files to the acquisition of aligned Procrustes coordinates (*i.e.* sound shape), it is now possible to proceed with the ordination and visualization of our data. In Geometric Morphometrics, the Principal Component Analysis (PCA) and the Canonical Variates Analysis (CVA) are the main ordination methods typically employed (Zelditch et al., 2012). Even though PCA and CVA are complementary analysis that simplify description of variation, they both serve different purposes. PCA is a ordination method that creates variables useful when examining variation among individuals, whereas CVA construct variables to describe the relative positions of groups within the sample (Zelditch et al., 2012). Although these methods do not represent tests of hypothesis *per se*, both result in ordinations that simplify description and are very useful in exploratory data analysis, specially when one is looking for potential groups within the data.

Among the benefits of employing a PCA on multivariate data is the possibility to generate two dimensional plots that encompass the majority of variation embedded in the data (Zelditch et al., 2012). This is done by calculating new variables (Principal Components – PC) that represent the axis of greatest variation within the data and are independent from each other. However it is desirable to employ a “stopping rule” to select which PCs should be retained as nontrivial and interpretable and which ones should be ignored. Jackson (1993) reviewed those methods and recommend the use of broken stick model based on eigenvalues (Frontier, 1976), which consists in the visual interpretation of relative variance from principal components (*i.e.* eigenvalues or inertia) in relation to the expected variance if data had a random distribution. The PCs retained by the broken stick criteria are the ones with eigenvalues (*i.e.* variation) higher than that from a random distribution, thus reducing the probability of selecting PCs that represent random data.

Although `stats` package feature `screeplot` function with a `bstick` argument for computing a broken stick model, this function does not work within `rmarkdown`. Therefore, we introduce new function `bsm` that also computes a broken stick model based on eigenvalues. `bsm` output is as a barplot of relative variance from the PCA eigenvalues, and the broken stick model as a red line (Figures 41 and 41). PCs whose variance is higher than predicted by the broken stick model are the ones to be retained in the PCA analysis.

R code: In order to use `bsm` function one must first import it into the R environment. As described for the `frequencies`, `ZC`, `SoundShape` and `eigensound` functions, this can be done by pasting the whole code below into R’s console and then executing it, or by sourcing the function into R (*i.e.* `source("insert.filepath.here/broken-stick-model.R")`).

Table 4: Arguments to be used by broken stick model `bsm` function.

Arguments	Description
<code>PCA.out</code>	The output of a Principal Components Analysis performed by the function <code>prcomp</code> .
<code>ylim</code>	Limits of Y-axis given as percentage of variance on a vector of two values between 0 and 100. By default, <code>bsm</code> will use the maximum variance from the PCs, however one might be interested in higher limits if <i>e.g.</i> broken stick model exceeds the highest variation from PCs.
<code>nPCs</code>	Number of Principal Components to display as barplot.
<code>main</code>	Main title of output plot.

```

# Function for the visualization of broken stick model and
percentage of variance in PCs

bsm <- function(PCA.out, main="Broken stick model",
ylim=c(0,max(ev.p)), nPCs=length(colnames(PCA.out$x))){
  ev = PCA.out$sdev^2 # Eigenvalues
  ev.p = ev/sum(ev)*100 # Eigenvalues given as percentage of
variation

  bsm = vegan::bstick(PCA.out) # Broken stick model
  bsm.p <- bsm/sum(bsm)*100 # Broken stick model given as
percentage of variation

  # Visualize broken stick as percentage of variation for each
Principal Components
  barp <- barplot(ev.p[1:nPCs], main=main, ylim=ylim, las=1,
names.arg = colnames(PCA.out$x[,1:nPCs]),
ylab="Percentage of variance")
  lines(x=barp[1:nPCs], y=bsm.p[1:nPCs], col="red")
  points(x=barp[1:nPCs], y=bsm.p[1:nPCs], col="red")
}

```

We now delve into the PCA performed with the Procrustes residual coordinates generated in previous code chunks. Our sample is composed of a single representative advertisement call from 30 *Physalaemus* species, thus comprising a wide variety of sound shapes to illustrate the behaviour of both *SoundShape* and *eigensound* methods. Each species received a number (see code chunk bellow) according to their relative position in the simplified phylogeny displayed in Figure 1 of Rocha & Romano (*in prep*). Numbers are not intended to provide any biological or evolutionary meaning, they are simply the visual arrange of shapes in the phylogenetic tree adapted from Lourenço et al. (2015).

R code: As commented above, the Generalized Procrustes Analysis (GPA) of semilandmark coordinates from *SoundShape* are hereafter conducted using the minimum Procrustes distance criterion for sliding semilandmarks along the curve of relative amplitude. Therefore, this analysis is repeated for organization purposes. A Principal Components Analysis (PCA) `prcomp` is performed using the Procrustes coordinates acquired in the GPA of both *SoundShape* and *eigensound* outputs, and also on the original semilandmark coordinates from *eigensound* (*i.e.* same approach

from MacLeod et al. 2013). Note that `prcomp` must be performed on a two dimensional array (*i.e.* `matrix`), so it is necessary to convert the three dimensional arrays of coordinates using `two.d.array` before PCA. Relative variances and broken stick model are plotted using new function `bsm`.

```
# Source bsm function into R's environment
source("insert.filepath.here/broken-stick-model.R")

# GPA with semilandmarks allowed to slide using Procrustes
distance criterion
SS.gpa.Proc <- gpagen(SS.sample, PrinAxes = FALSE,
print.progress = FALSE, curves = sliders.SS, ProcD = TRUE)

# Principal Components Analysis (PCA) using Procrustes
coordinates from SoundShape
SS.pca.sample <- prcomp(two.d.array(SS.gpa.Proc$coords))

# Broken stick model for relative variances in each PCs
bsm(SS.pca.sample, main="Broken stick model - PCA SoundShape")
```

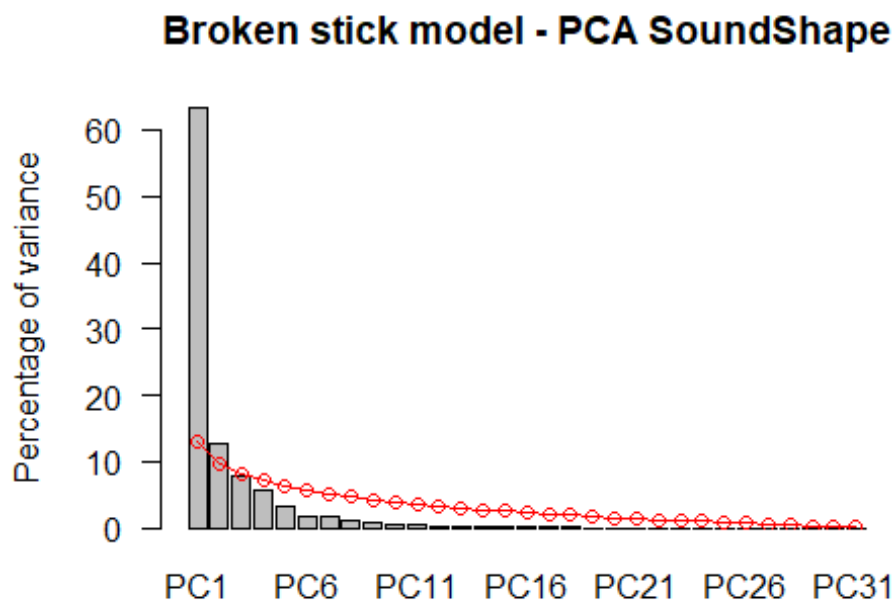


Figure 40: Eigenvalues as percentage of variance (grey bars) and broken stick model (red dots and line) computed from Principal Components analysis using Procrustes coordinates. Data acquired using SoundShape function and 2D residuals after GPA + sliding.

```

# Perform Generalized Procrustes Analysis on semilandmarks
acquired by eigensound
eig.gpa<-gpagen(eig.sample,PrinAxes=FALSE,print.progress=FALSE)

# Principal Components Analysis (PCA) using semilandmark
coordinates from eigensound
eigSM.pca.sample <- prcomp(two.d.array(eig.sample))

# Principal Components Analysis (PCA) using Procrustes
coordinates from eigensound
eig.pca.sample <- prcomp(two.d.array(eig.gpa$coords))

# Broken stick model for relative variances in each PCs
par(mfrow=c(1,2))
bsm(eigSM.pca.sample, ylim=c(0,14), main="Broken stick model -
eigensound coordinates")
bsm(eig.pca.sample, ylim=c(0,14), main="Broken stick model -
eigensound after GPA")

```

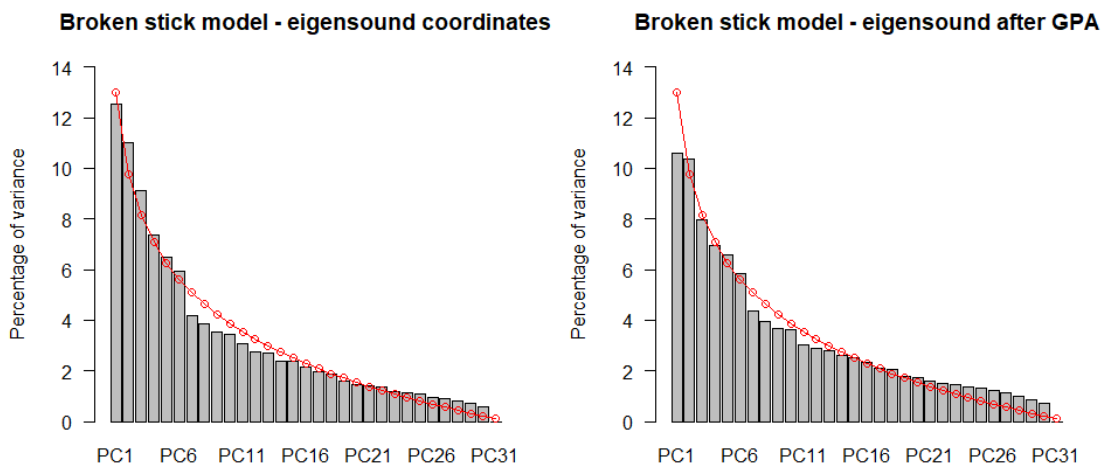


Figure 41: Eigenvalues as percentage of variance (grey bars) and broken stick model (red dots and line) computed from Principal Components analysis using (right) original coordinates acquired by eigensound function (left) and Procrustes coordinates after GPA (right).

The visual inspection of broken stick models (Figures 40 and 41) allowed us to select only nontrivial and interpretable principal components based on the relative variance of eigenvalues against the broken stick random distribution. For the SoundShape analysis, this criteria led to the unambiguous selection of PC1 (63.3% variance) and PC2 (12.8% variance), which summed up to almost 77% of the whole variance in our data (Figures 40 and 42).

On the other hand, the results from the `eigensound` analysis were inconclusive. Regardless of the use of semilandmark or Procrustes coordinates, the relative variance of eigenvalues is close to that of a random distribution of data from the broken stick model (Figure 41). This indicate that the data gathered through the *eigensound* approach resulted in such a large amount of variation that it is not feasible to summarize it into a few principal components, even after performing a Generalized Procrustes Analysis. In addition, the first PC comprised a little over 12% variation using original semilandmark coordinates, and a little over 10% variation using Procrustes coordinates after GPA, which is less than predicted for a random distribution (Figure 41) and enough to disapprove the use of PCA as a feasible ordination approach in this case.

In spite of that, we will proceed with the visualization of the first two principal components similarly as the analysis performed by MacLeod et al. (2013) (these authors also provided plots that included the third PC). However we will use the Procrustes coordinates rather than the original semilandmark coordinates in attempt to deal with data that had a noise reduction.

To facilitate the subsequent plots of Principal Components, we introduce new function `pca.plot`, which require the result `PCA.out` of a PCA performed with `prcomp`, a vector with `groups`, a vector of colours `col.gp` intended for each group and for each convex hull `col.conv`. In addition, it is also possible to define which Principal Components PCs will be plotted and the `main` title for the plot.

R code: In order to use `pca.plot` function one must first import it into the R environment. As described for the previous functions (*e.g* `SoundShape`, `eigensound`, `bsm`), this can be done by pasting the whole code below into R's console and then executing it, or by sourcing the function into R (*i.e.* `source("insert.filepath.here/pca-plot.R")`).

Table 5: Arguments to be used by `plot.pca` function.

Arguments	Description
<code>PCA.out</code>	The output of a Principal Components Analysis performed by the function <code>prcomp</code> .
<code>groups</code>	Groups to use as colors and/or convex hulls. A vector with the same length as the number of rows from coordinates of principal components in the PCA output <code>nrow(PCA.out\$x)</code> .
<code>conv.hulls</code>	Groups to use for convex hulls. A vector with the same length as the number of rows from coordinates of principal components in the PCA output <code>nrow(PCA.out\$x)</code> . Defalut is plot without convex hulls.

<code>sp.as</code>	Enables one to choose between plotting elements as "points" (default) or "text". If <code>sp.as="text"</code> , then one must also input a vector of characters to use (see argument <code>sp.text</code>).
<code>sp.text</code>	Elements as text to be plotted. A vector with the same length as the number of rows from coordinates of principal components in the PCA output <code>nrow(PCA.out\$x)</code> .
<code>PCs</code>	Intended Principal Components to plot. A vector of length two.
<code>col.gp</code>	A vector of colours for the intended groups. Must be the same length as the number of group levels <code>levels(groups)</code> . By default, <code>pca.plot</code> will create a vector of colors that has the same length of levels in groups.
<code>col.conv</code>	A vector of colours for the intended convex hulls <code>conv.hulls</code> . Must be the same length as the number of convex hulls <code>levels(conv.hulls)</code> . Default is plot without convex hulls.
<code>main</code>	Main title of output plot.
<code>leg</code>	A logical. If TRUE, a legend is added to plot.
<code>leg.labels</code>	Labels to use when <code>leg=TRUE</code> . Must be the same length of levels in groups. By default, it will use as labels the levels in groups.
<code>leg.position</code>	Specify legend location using a keyword from the list "bottomright", "bottom", "bottomleft", "left", "topleft", "top", "topright", "right" and "center". Alternatively, a single value can be provided and used for both margins, or two values, in which the first is used for X distance, and the second for Y distance. See also <code>?legend</code> for details.

```
# Function for plotting ordination of Principal Components
pca.plot <- function(PCA.out, groups, conv.hulls=NULL,
sp.as="points", sp.text, PCs=c(1,2),
col.gp=rainbow(length(levels(groups))),
col.conv=rainbow(length(levels(conv.hulls))),
main="PCA of Procrustes coordinates", leg=FALSE,
leg.labels=groups, leg.pos="topright")
{
  # Store PC scores as new object
  PCscores <- summary(PCA.out)$importance

  # Use PC scores as labels in PCA visualization
  xlab = paste("Principal Component ", as.character(PCs[1]),
" (", round(PCscores[2,PCs[1]]*100, 1), "%)", sep="")
  ylab = paste("Principal Component ", as.character(PCs[2]),
" (", round(PCscores[2,PCs[2]]*100, 1), "%)", sep="")

  # create a colour vector to colour the groups
  names(col.gp) = levels(groups) # dimension labels
```

```

col.groups = col.gp[match(groups, names(col.gp))] # assign a
colour to each specimen

#Principal components species distribution
plot(PCA.out$x[,PCs[1]], PCA.out$x[,PCs[2]], type='n', las=1,
xlab=xlab, ylab=ylab, main=main, cex.axis=1.1, cex.lab=1.55,
cex.main=1.8)

# Add convex hulls
if(!is.null(conv.hulls)){ for(j in 1:nlevels(conv.hulls)) {
  # Get edge points (used to plot convex hull):
  edge_points = rownames(PCA.out$x[which(conv.hulls ==
levels(conv.hulls)[j]),)] [chull(PCA.out$x[which(conv.hulls ==
levels(conv.hulls)[j]), PCs)])
  # Plot convex hull as polygon:
  polygon(PCA.out$x[edge_points, PCs], col =
adjustcolor(col.conv[j], alpha.f = 0.1), border = col.conv[j])}
  if(sp.as=="points"){points(PCA.out$x[,PCs[1]],
PCA.out$x[,PCs[2]], pch=21, cex=1.5, bg=col.groups)}
  if(sp.as=="text"){text(PCA.out$x[,PCs[1]], PCA.out$x[,PCs[2]],
as.factor(sp.text), col=col.groups, cex=1.4)}

# Add legend
if(leg==TRUE){legend(x=leg.pos, legend=unique(leg.labels),
pch=21, pt.bg=unique(col.groups))}
}

# Source pca.plot function into R's environment
source("insert.filepath.here/pca-plot.R")

```

R code: The code chunk below focus on the construction of a `data.frame` to use as groups in the subsequent PCA plots. Groups follow the proposal from Lourenço et al. 2015. Each object created represent a column in the `data.frame`. Ordering of species follow that of Figure 1 from Rocha & Romano (*in prep*).

```

# Create data frame with groups to use in the analysis. Groups
according to Lourenço et al. 2015.
nums <- as.factor(seq(1, 30, 1))
spp <- c("P. nattereri", "P. maculiventris", "P. camacan",

```

```
"P. crombiei", "P.obtectus", "P. signifer", "P. moreirae",
"P. bokermanni", "P. nanus", "P. spiniger", "P. rupestris",
"P. deimaticus", "P. erythros", "P. cicada", "P. feioi",
"P. olfersii", "P. soaresi", "P. lisei", "P. gracilis",
"P. evangelistai", "P. marmoratus", "P. biligonigerus",
"P. riograndensis", "P. aguirrei", "P. albonotatus", "P. kroyeri",
"P. albifrons", "P. centralis", "P. cuvieri", "P. fischeri")
```

```
group <- c(rep("Stem P. cl. signifer", 2),
rep("P. gr. signifer", 8), rep("P. gr. deimaticus", 3),
"Stem P. cl. cuvieri", rep("P. gr. olfersii", 3),
rep("P. gr. gracilis", 3), rep("P. gr. biligonigerus", 3),
"Stem P. cl. cuvieri", rep("P. gr. cuvieri", 7))
clade<-c(rep("P. cl. signifer", 13), rep("P. cl. cuvieri", 18))
```

```
Physa <- data.frame(nums, spp, group, clade)
```

```
Physa
```

##	nums	spp	group	clade
## 1	1	P. nattereri	Stem P. cl. signifer	P. cl. signifer
## 2	2	P. maculiventris	Stem P. cl. signifer	P. cl. signifer
## 3	3	P. camacan	P. gr. signifer	P. cl. signifer
## 4	4	P. crombiei	P. gr. signifer	P. cl. signifer
## 5	5	P.obtectus	P. gr. signifer	P. cl. signifer
## 6	6	P. signifer	P. gr. signifer	P. cl. signifer
## 7	7	P. moreirae	P. gr. signifer	P. cl. signifer
## 8	8	P. bokermanni	P. gr. signifer	P. cl. signifer
## 9	9	P. nanus	P. gr. signifer	P. cl. signifer
## 10	10	P. spiniger	P. gr. signifer	P. cl. signifer
## 11	11	P. rupestris	P. gr. deimaticus	P. cl. signifer
## 12	12	P. deimaticus	P. gr. deimaticus	P. cl. signifer
## 13	13	P. erythros	P. gr. deimaticus	P. cl. signifer
## 14	14	P. cicada	Stem P. cl. cuvieri	P. cl. cuvieri
## 15	15	P. feioi	P. gr. olfersii	P. cl. cuvieri
## 16	16	P. olfersii	P. gr. olfersii	P. cl. cuvieri
## 17	17	P. soaresi	P. gr. olfersii	P. cl. cuvieri
## 18	18	P. lisei	P. gr. gracilis	P. cl. cuvieri
## 19	19	P. gracilis	P. gr. gracilis	P. cl. cuvieri
## 20	20	P. evangelistai	P. gr. gracilis	P. cl. cuvieri

```
## 21 21 P. marmoratus P. gr. biligonigerus P. cl. cuvieri
## 22 22 P. biligonigerus P. gr. biligonigerus P. cl. cuvieri
## 23 23 P. riograndensis P. gr. biligonigerus P. cl. cuvieri
## 24 24 P. aguirrei Stem P. cl. cuvieri P. cl. cuvieri
## 25 25 P. albonotatus P. gr. cuvieri P. cl. cuvieri
## 26 26 P. kroyeri P. gr. cuvieri P. cl. cuvieri
## 27 27 P. albifrons P. gr. cuvieri P. cl. cuvieri
## 28 28 P. centralis P. gr. cuvieri P. cl. cuvieri
## 29 29 P. cuvieri P. gr. cuvieri P. cl. cuvieri
## 30 30 P. fischeri P. gr. cuvieri P. cl. cuvieri
```

R code: Principal Components are plotted with the function `pca.plot` as introduced above. Legend is plotted separately from image file.

```
# Load legend as jpeg image and store as R object
leg.Physa.orig <- imager::load.image("C:/R/Physalaemus-Sample/
legend-pca-Physa.jpg")
leg.Physa <- imager::crop.borders(leg.Physa.orig, nx=90, ny=70)

# Plot PCA coordinates from SoundShape approach
layout(matrix(c(1,1,2), 1, 3))
par(mgp=c(3,0.7,0))
pca.plot(PCA.out=SS.pca.sample, groups=Physa$group,
conv.hulls = Physa$clade, sp.as="text", sp.text=Physa$nums,
PCs=c(1,2), col.gp=c("#6666CC", "#66CC33", "#FF0000", "#0000FF",
"#CC66FF", "#FF6600", "#000000", "#000000"), col.conv=c("green3",
"yellow2"), main="SoundShape - PCA of Procrustes coordinates")
plot(leg.Physa, axes=F) # Legend
```

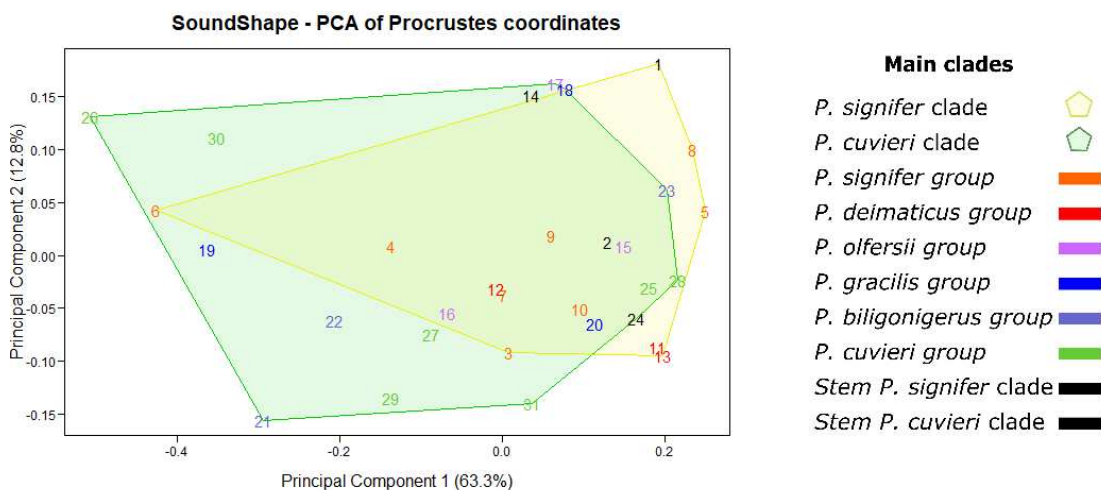


Figure 42: Individual projection of 30 *Physalaemus* species onto PC1 and PC2 axes using Procrustes

coordinates acquired by SoundShape function. Numbers are colored according to phylogenetic groups from Lourenço et al. (2015) and convex hulls encompass major clades within *Physalaemus* (for numbers relative to species, groups and clades, see previous code chunk and Rocha & Romano, *in prep*, Fig. 1). Legend represent colors for each *Physalaemus* groups and clades.

```
# Plot PCA coordinates from eigensound approach
layout(matrix(c(1,1,2), 1, 3))
par(mgp=c(3,0.7,0))
pca.plot(PCA.out=eig.pca.sample, groups=Phy$a$group,
conv.hulls=Phy$a$clade, sp.as="text", sp.text=Phy$a$num,
PCs=c(1,2), col.gp=c("#6666CC", "#66CC33", "#FF0000", "#0000FF",
"#CC66FF", "#FF6600", "#000000", "#000000"), col.conv=c("green3",
"yellow2"), main="eigensound - PCA of Procrustes coordinates")
plot(leg.Phy$a, axes=F) # Legend
```

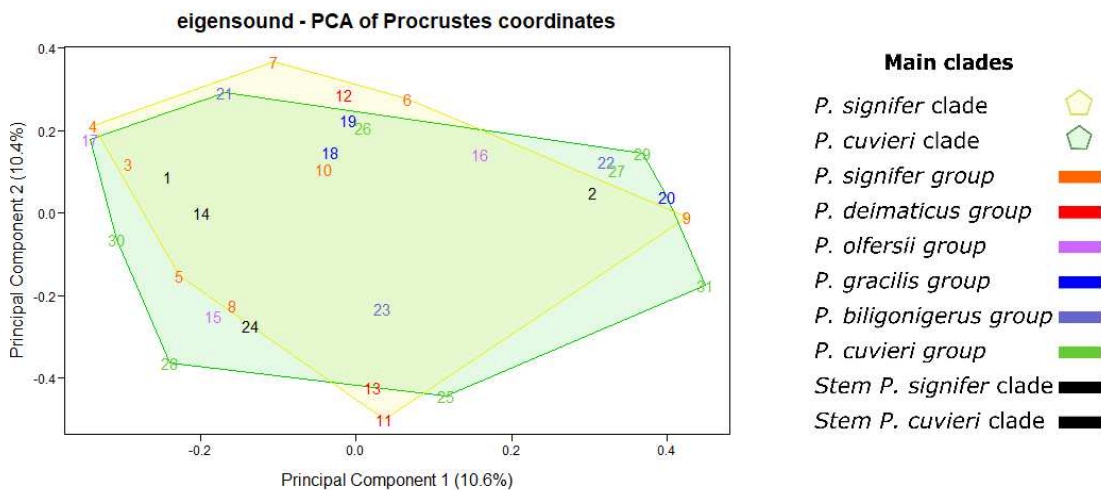


Figure 43: Individual projection of 30 *Physalaemus* species onto PC1 and PC2 axes using Procrustes coordinates acquired by eigensound function. Numbers are colored according to phylogenetic groups from Lourenço et al. (2015) and convex hulls encompass major clades within *Physalaemus* (for numbers relative to species, groups and clades, see previous code chunk and Rocha & Romano, *in prep*, Fig. 1). Legend represent colors for each *Physalaemus* groups and clades.

It is somewhat expected that the plots from figures 42 and 43 might seem a bit confusing. The PCAs encompass a large variety of sound shapes, each one represented by a single stereotyped advertisement call from 30 *Physalaemus* species. Moreover, as shown in Figure 1 from Rocha & Romano (*in prep*), the sound shapes of *Physalaemus* species have overall similarities that do not follow phylogenetic relationships among groups. Therefore, it would be surprising if the PCA analysis somehow managed to structure a clear cut grouping of species based on such variable dataset.

Thus, these two general properties of the data – large variation and convergent sound shapes – would explain the lack of structure in the ordination plot of the 30 *Physalaemus* species, at least taking into account the phylogenetic relationships as the

criterea to lable *a priori* groups. This is in agreement with the results from Cannatella et al. (1998), in which these authors found a low phylogenetic signal in the advertisement calls of *Engystomops pustulosus* group. However, this is enough to exemplify the application of the above mentioned methods and bound the sub-sample of subsequent analyses.

Ultimately, we can visualize the deformation grids (thin plates splines) as representations of how shape change along the axis of a principal component analysis in relation to consensus configuration (Zelditch et al. 2012). Although the function `plotTangentSpace` from `geomorph` also calculate hypothetical semilandmark configurations for each principal component, we introduce here the new function `PCshapes`, which focus only on the calculation of hypothetical semilandmark configurations for each Principal Component. `PCshapes` represent a simplification of `plotTangentSpace` as it does not feature as many functionalities as the original code. Hypothetical semilandmark configurations are calculated from Procrustes coordinates acquired by `gpagen` function.

The output of `PCshapes` must be assigned to an R object and features a list of hypothetical semilandmark configurations for minimum and maximum vales of each Principal Component. Configurations from PCs are used to generate deformation grids in relation to consensus shape using `plotRefToTarget`. Unfortunately, it is very demanding computationally to produce three dimensional deformation grids due to the large amount of data gathered by `eigensound` function. Therefore, we only provide hypothetical semilandmark configurations from PCs of the *eigensound* approach.

R code: In order to use `PCshapes` function one must first import it into the R environment. As described for the previous functions (*e.g* `SoundShape`, `eigensound`, `pca.plot`), this can be done by pasting the whole code below into R's console and then executing it, or by sourcing the function into R (*i.e.* `source("insert.filepath.here/PCshapes.R")`).

Table 6: Argument to be used by `PCshapes` function.

Arguments	Description
<code>GPA.out</code>	The output of a Generalized Procrustes Analysis performed by the function <code>gpagen</code> .

```
# Function PCshapes for computing shapes from Procrustes
coordinates (adapted from plotTangentSpace)
```

```

PCshapes <- function(GPA.out){

  # Principal Components Analysis (PCA) using Procrustes
coordinates

  PCA.out = prcomp(two.d.array(GPA.out$coords))

  pcddata = PCA.out$x
  ref = mshape(GPA.out$coords)
  p = dim(GPA.out$coords)[1]
  k = dim(GPA.out$coords)[2]

  shapes <- shape.names <- NULL
  for (i in 1:ncol(pcddata)) {
    pcaxis.min <- min(pcddata[, i])
    pcaxis.max <- max(pcddata[, i])
    pc.min <- pc.max <- rep(0, dim(pcddata)[2])
    pc.min[i] <- pcaxis.min
    pc.max[i] <- pcaxis.max
    pc.min <- as.matrix(pc.min %*% (t(PCA.out$rotation))) +
      as.vector(t(ref))
    pc.max <- as.matrix(pc.max %*% (t(PCA.out$rotation))) +
      as.vector(t(ref))
    shapes <- rbind(shapes, pc.min, pc.max)
    shape.names <- c(shape.names, paste("PC", i, "min",
sep = ""), paste("PC", i, "max", sep = ""))    }
    shapes <- arrayspecs(shapes, p, k)
    shapes <- lapply(seq(dim(shapes)[3]), function(x) shapes[, ,x])
    names(shapes) <- shape.names

  out <- shapes
}

# Source PCshapes function into R's environment
source("insert.filepath.here/PCshapes.R")

```

R code: Semilandmark configurations are acquired from Procrustes coordinates using PCshapes function. Deformation grids are plotted using plotRefToTarget for SoundShape data. Hypothetical three dimensional semilandmark configurations are plotted with scatter3D for eigensound semilandmarks.

```

# Acquire hypothetical semilandmark configurations from
Procrustes coordinates

SS.shapes <- PCshapes(SS.gpa.Proc)
eig.shapes <- PCshapes(eig.gpa)

# Deformation grids from Principal Components of SoundShape
analysis

# Principal Component 1
par(mar=c(1, 1, 1, 1), mfrow=c(1,2))
plotRefToTarget(SS.gpa.Proc$consensus, SS.shapes$PC1min)
title(main="PC1 - Minimum")
plotRefToTarget(SS.gpa.Proc$consensus, SS.shapes$PC1max)
title(main="PC1 - Maximum")

```

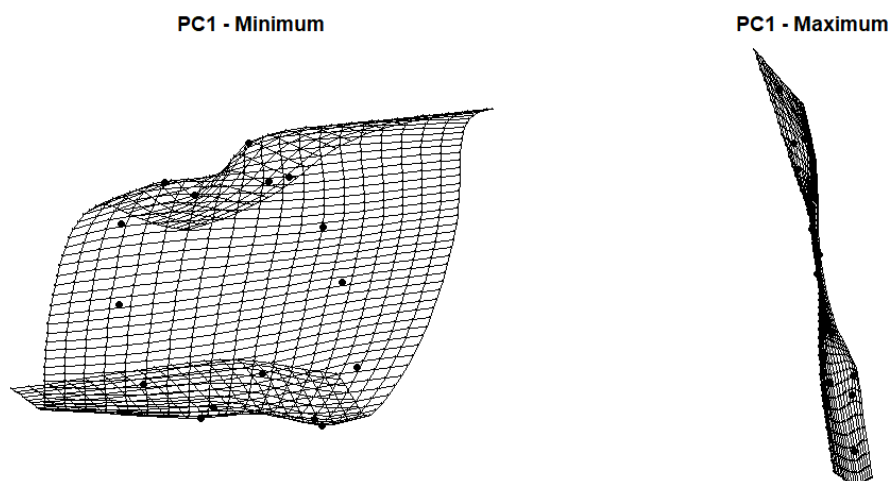


Figure 44: Semilandmarks (dots) and deformation grids of minimum (left) and maximum (right) scores of the first Principal Component (PC1; 62.9% of variance). Deformation grids are given in relation to consensus configuration (i.e. score 0 of PC1) calculated from Procrustes coordinates acquired by SoundShape function. Data gathered from a representative advertisement call of 30 *Physalaemus* species.

```

# Principal Component 2
par(mar=c(1, 1, 1, 1), mfrow=c(1,2))
plotRefToTarget(SS.gpa.Proc$consensus, SS.shapes$PC2min)
title(main="PC2 - Minimum")
plotRefToTarget(SS.gpa.Proc$consensus, SS.shapes$PC2max)
title(main="PC2 - Maximum")

```

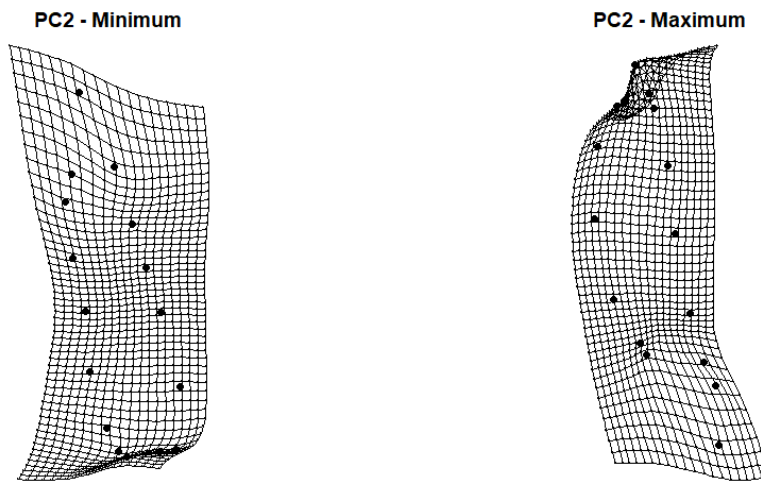


Figure 45: Semilandmarks (dots) and deformation grids of minimum (left) and maximum (right) scores of the second Principal Component (PC2; 14% of variance). Deformation grids are given in relation to consensus configuration (i.e. score 0 of PC2) calculated from Procrustes coordinates acquired by SoundShape function. Data gathered from a representative advertisement call of 30 *Physalaemus* species.

```
# Hypothetical three dimensional semilandmark configurations
from PCs of eigensound analysis

# Principal Component 1
par(mfrow=c(1,2))

scatter3D(x=eig.shapes$PC1min[, 1], y=eig.shapes$PC1min[, 2],
z=eig.shapes$PC1min[,3], theta=30, phi=35, resfac=1, r=3,
expand=0.5, scale=T, axes=T, cex.axis=0.7, ticktype="detailed",
nticks=4, pch=21, col=spectro.colors(n=100), cex=0.4, xlab="Time
coordinates", ylab="Frequency coordinates",
zlab="Amplitude coordinates", clab=expression('Amplitude'),
main=expression('PC1 - Minimum'))

scatter3D(x=eig.shapes$PC1max[, 1], y=eig.shapes$PC1max[, 2],
z=eig.shapes$PC1max[, 3], theta=30, phi=35, resfac=1, r=3,
expand=0.5, scale=T, axes=T, cex.axis=0.7, ticktype="detailed",
nticks=4, pch=21, col=spectro.colors(n=100), cex=0.4,
xlab="Time coordinates", ylab="Frequency coordinates",
zlab="Amplitude coordinates", clab=expression('Amplitude'),
main=expression('PC1 - Maximum'))
```

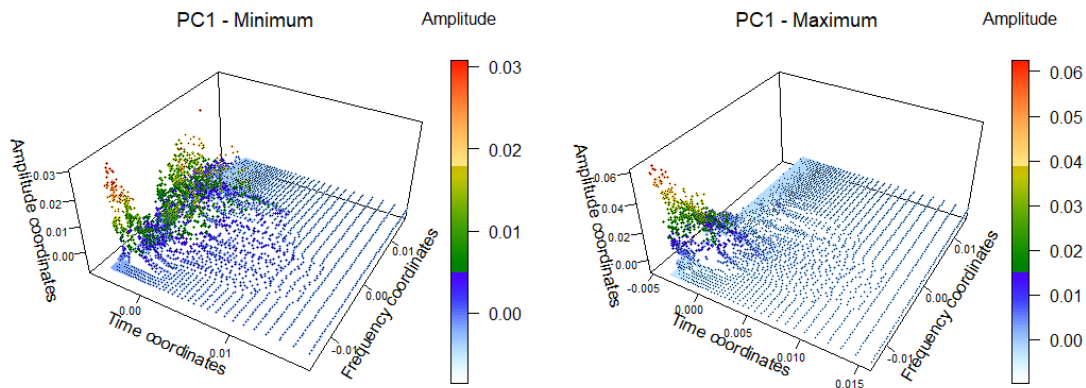


Figure 46: Hypothetical semilandmark configurations of minimum (left) and maximum (right) scores of the first Principal Component (PC1; 10.6% of variance) calculated from Procrustes coordinates acquired by eigensound function. Data gathered from a representative advertisement call of 30 *Physalaemus* species.

```
# Principal Component 2
par(mfrow=c(1,2))
scatter3D(x=eig.shapes$PC2min[, 1], y=eig.shapes$PC2min[, 2],
z=eig.shapes$PC2min[, 3], theta=30, phi=35, resfac=1, r=3,
expand=0.5, scale=T, axes=T, cex.axis=0.7, ticktype="detailed",
nticks=4, pch=21, col=spectro.colors(n=100), cex=0.4,
xlab="Time coordinates", ylab="Frequency coordinates",
zlab="Amplitude coordinates", clab=expression('Amplitude'),
main=expression('PC2 - Minimum'))
scatter3D(x=eig.shapes$PC2max[, 1], y=eig.shapes$PC2max[, 2],
z=eig.shapes$PC2max[, 3], theta=30, phi=35, resfac=1, r=3,
expand=0.5, scale=T, axes=T, cex.axis=0.7, ticktype="detailed",
nticks=4, pch=21, col=spectro.colors(n=100), cex=0.4,
xlab="Time coordinates", ylab="Frequency coordinates",
zlab="Amplitude coordinates", clab=expression('Amplitude'),
main=expression('PC2 - Maximum'))
```

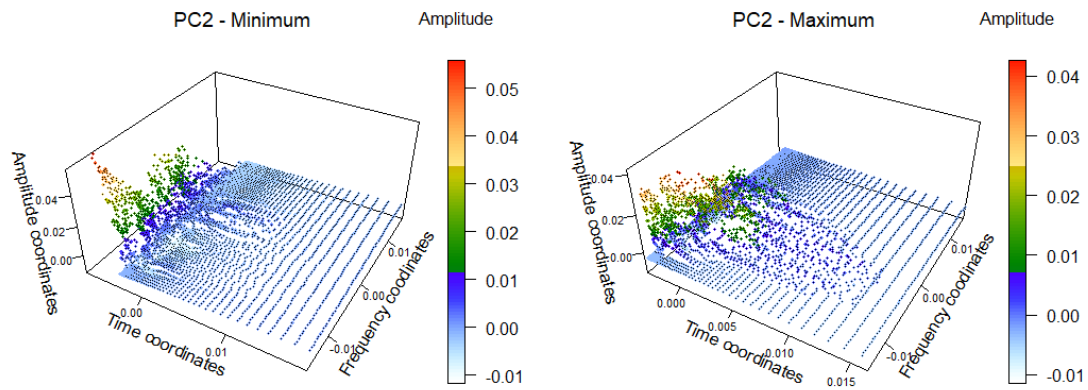


Figure 47: Hypothetical semilandmark configurations of minimum (left) and maximum (right) scores of the second Principal Component (PC2; 10.4% of variance) calculated from Procrustes coordinates acquired by `eigenSound` function. Data gathered from a representative advertisement call of 30 *Physalaemus* species.

Deformation grids (Figures 44 and 44), hypothetical semilandmark configurations (Figures 46 and 46) and Principal Components (Figures 42 and 43) must be analyzed together in order to fully comprehend changes in sound shape along a sample.

In the example of `SoundShape` data above, we used a broken stick criteria to retain PC1 and PC2, in which PC1 encompass 63.3% of total variation and PC2 encompass 12.8% (Figure 40). Therefore, the resulting plot of Principal Components represent 76.1% of the whole variation embedded in our dataset (Figure 42). This great variation is expressed as great deformation in the thin plates splines (Figures 44 and 45), with higher PC1 values representing very slim call shapes (e.g. 1 - *P. nattereri*, 5 - *P. obtectus*, 8 - *P. bokermanni*, 11 - *P. rupestris*, 13 - *P. erythros*) and lower PC1 values representing broad call shapes (e.g. 6 - *P. signifer*, 19 - *P. gracilis*, 26 - *P. kroyeri*, 30 - *P. cuvieri*). The PC2 only represent 12.9% variation, and thus the grids from this axis presented less dramatic deformations, with subtle differences in calls displayed as extreme values in our PCA plot (Figure 42).

Since we only analyzed a single representative note from 30 *Physalaemus* species, the lack of clear structure in our PCA is no surprise and should not be taken on a high note. Important to notice that `SoundShape` approach was insensitive to the call structure from each analyzed sound, meaning that the shapes from harmonic and non-harmonic calls did not present distinct PC scores. Also worth mentioning that main differences in call shape cannot be explained by phylogenetic distances among major *Physalaemus* groups (*sensu* Lourenço et al. 2015), as 2D shape patterns do not appear to be a trend when looking at several biological groups at once. However these are all very premature

conclusions made from a limited sample, in which our objective was mainly methodological.

Regarding the *eigensound* approach, our broken stick criteria suggests that neither PCs should be considered as nontrivial and retained in our analysis (Figure 41). In fact, the variation encompassed by the Principal Components follow the same trend expected for a data randomly distributed (Jackson, 1993). Even though the combined variation explained by PC1 and PC2 represent only 21% of the whole variation in our data, we proceeded with the ordination plot (Figure 43) as well as hypothetical semilandmark configurations (Figures 46 and 47) in order to comprise the whole *eigensound* methods described by MacLeod et al. (2013) and modified here.

In reality, MacLeod et al. (2013) achieved similar results in their *eigensound* analysis. In addition to the low variation encompassed by the PCA plots from these authors (*i.e.* less than 40%, combining PC1 and PC2), their ordination plots resulted in a *horseshoe* pattern of distribution. Although the horseshoe phenomenon is not ideal (*i.e.* it indicates non-linearity in data, but can be caused by a variety of data distributions; Podani & Miklós, 2002), it is not prohibitive when interpreting a Principal Component Analysis (see discussion in Podani & Miklós, 2002; and MacLeod et al., 2013). In light of the low statistical power yielded by our results, added to the already reported misbehaviour of data extracted from the three dimensional surface of sound (see MacLeod et al., 2013), we withdrawn from further discussing the results from these methods up to this point. Any conclusions drawn from our data sample incur the error of treating random data as representative.

Concluding remarks

We have now concluded the step by step guide to the sound shape analysis employing tools extracted from traditional bioacoustics and from Geometric Morphometrics. Thus far, we focused on exploratory analysis from *SoundShape* (Rocha & Romano, *in prep*) and *eigensound* approaches (MacLeod et al., 2013), aiming to provide tools for the methods along with a glimpse on the theoretical background for anyone to replicate and implement these methods on R platform. Nevertheless, Geometric Morphometrics is a broad and expanding discipline whose methods go well beyond the topics addressed in

this supplementary file. Further studies might shed light on other GM approaches towards understanding sound shape from either two or three dimensional sonic coordinates.

Although the codes provided in this supplementary material could be intimidating for those unfamiliar with R platform, the next and final section will focus on providing clear guidance for anyone to conduct sound shape studies on any taxonomical level. Please note that the codes provided herein are still under development and errors are likely to appear in the near future.

In addition, despite the low statistical power yielded by the *eigensound* approach, we will proceed with further comparisons of both methods as we dig into the various taxonomical levels within *Physalaemus*.

The shape of sound within *Physalaemus* (Anura: Leptodactylidae): An empirical study employing SoundShape and eigensound functions

In this final section, we will address two basic questions on sound shape variation among the genus *Physalaemus*. First, we verified if both SoundShape and eigensound methods are able to discriminate between the advertisement calls of eight *Physalaemus* species. Then, we address the interpopulational variation in the call of three populations of *P. deimaticus*.

All recordings employed in this work are available from bioacoustical collections (for a complete list of recordings, see Appendix 1 from Rocha & Romano *in prep*).

Note: The results of these empirical studies are also discussed in Rocha & Romano (*in prep*).

The sound shape between *Physalaemus* species

The first empirical study of sound shape focused on eight *Physalaemus* species selected from the sample of 30 species employed on the previous sections of this supplementary file (see Figures 19 and 34; see also Figure 1 in Rocha & Romano, *in prep*, for phylogenetic relations between species). Selection encompassed species belonging to the *P. signifer* clade (*sensu* Lourenço et al. 2015) that share similar spectrogram outputs (*e.g.* *P. nattereri* and *P. obtectus*; *P. signifer* and *P. spiniger*); a monophyletic species group whose species share similar advertisement calls (*i.e.* *P. deimaticus* group, including *P. deimaticus*, *P. erythros* and *P. rupestris*); and a species from the *P. cuvieri* clade (*i.e.* *P. cuvieri*). Therefore, this empirical study will measure the potential application of both SoundShape and eigensound functions when one wishes to discriminate species based on their call shapes, which can either be clearly different (*e.g.* *P. cuvieri* and *P. rupestris*) or remarkably similar (*e.g.* *P. deimaticus* and *P. erythros*).

Aside from *Physalaemus obtectus*, which had only a single sampled individual, each species had three individuals selected according to call availability from bioacoustical collections (see Appendix 1 from Rocha & Romano, *in prep*, for a complete list of recordings). In addition, each individual had five stereotyped notes selected

according to optimal signal to noise ratio and no overlap with background individuals. Therefore, our interspecific study encompassed 105 notes from 21 individuals belonging to eight *Physalaemus* species. Each note is stored as a separate wave file and placed at approximately time = 0, with remaining of sound window completed with silence. Sound windows were set to range from 0 to 0.5 seconds and from 0 to 5 kHz, thus encompassing long and short notes within the sample (Figure 48). Spectrogram parameters were set with window length `wl=512`, sampling rate `f=44100` and 80% overlap `ovlp=80` for increased resolution on time X-axis.

For illustrative purposes, `SoundShape` and `eigensound` analysis will be run separately in this first empirical study.

SoundShape analysis – In addition to the acquisition of semilandmark coordinates (SM) from the curve of relative amplitude, `SoundShape` function can also be useful on the exploratory phases of sound shape analysis. Since both 2D and 3D approaches require the manual placement of sounds at time = 0 s, by enabling `SoundShape` to export spectrogram outputs as `jpeg` files, one can easily verify the placement of notes on the sound window. More importantly, `SoundShape` analysis made *a priori* serve as a protocol for error verification, in the sense that the `jpeg` images display how semilandmarks are placed along the curve of relative amplitude for each individual and species within the study sample. This is an important step in sound shape analysis as it prevents the comparison of shapes altered by sonic artifacts (*e.g.* background noise) or overlapping sounds from multiple individuals.

R code: `SoundShape` function is run without storing the semilandmark coordinates. This is done as a protocol for error verification in which `jpeg` images are allowed to be exported by calling the argument `plot.exp=TRUE` and stating a filepath `store.at` where spectrogram outputs will be stored.

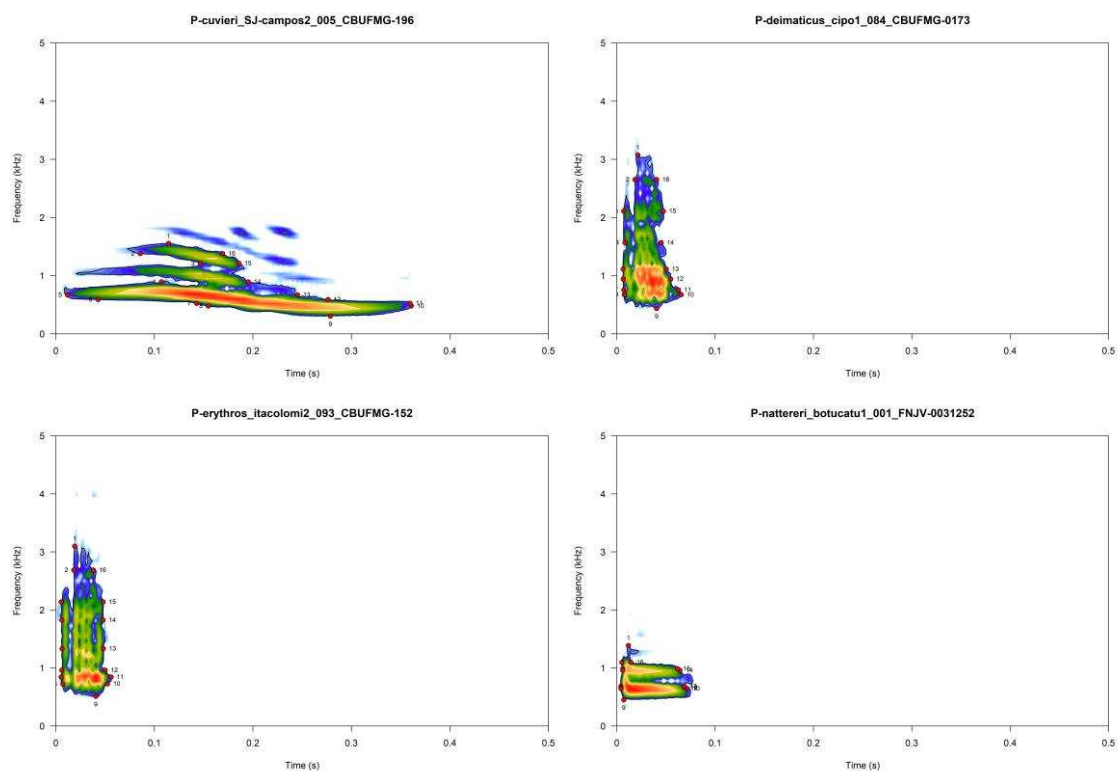
```
# Run SoundShape analysis to define range of sound window and  
place sounds at time=0  
  
SoundShape(tlim=c(0,0.5), flim=c(0,5), ovlp=80, plot.exp=TRUE,  
store.at = "C:/R/Physalaemus - interspecific/SoundShape",  
wav.at= "C:/R/Physalaemus - interspecific")
```

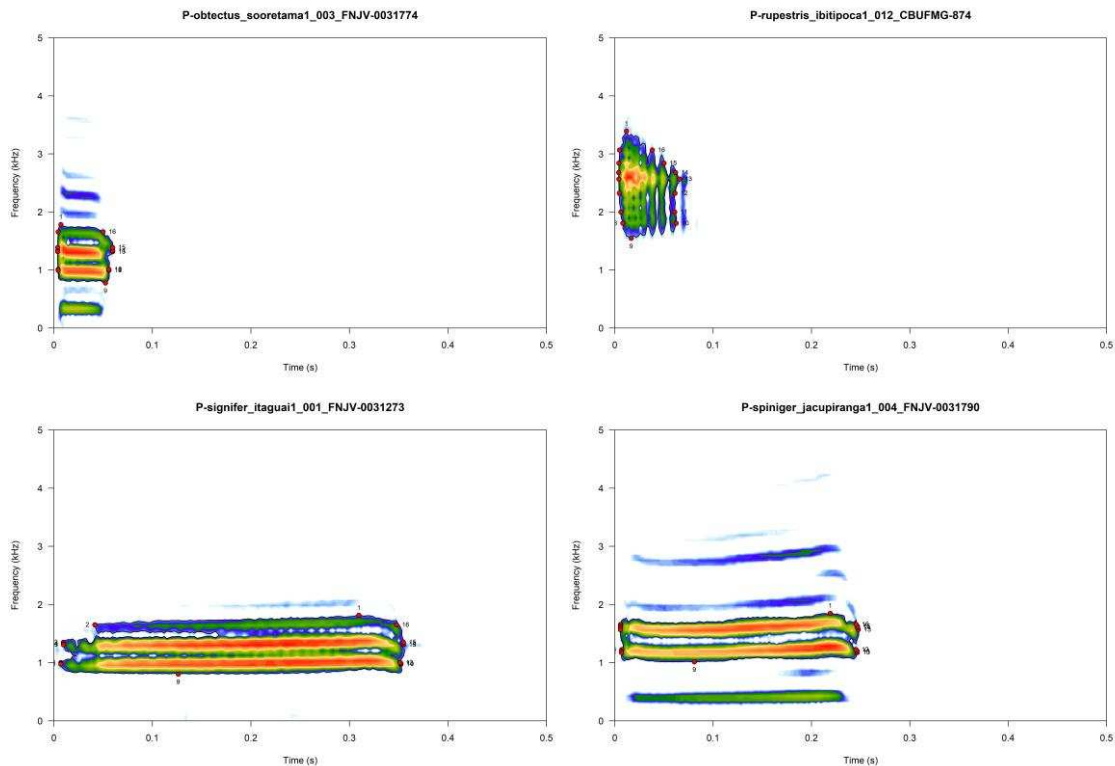
Once the spectrogram outputs had been verified for correct sound placement at time = 0, with appropriate acquisition of semilandmarks from each curve of relative

amplitude at -25 dB, we can use the jpeg images created by SoundShape as representative plots of spectrograms for each species in our sample (Figure 48).

Figure 48: (Displayed below) Spectrogram views of the advertisement calls of eight *Physalaemus* species including the set of 16 semilandmarks acquired from the curve of relative amplitude at -25 dB. Each species had a representative note selected previously. Spectrograms were generated automatically by SoundShape function.

```
# Display spectrograms from folder
include_graphics(list.files("C:/R/Physalaemus - interspecific/
sample", pattern=".jpg", full.names=TRUE))
```





After the verification of semilandmark acquisition by `SoundShape` function, all notes selected are stored in a separate folder from which analysis will be conducted. `SoundShape` is ran under its default settings, which yields the acquisition of 16 semilandmarks from the curve of relative amplitude at -25 dB, with each SM representing an energy quantile within the curve.

R code: `SoundShape` is ran mainly with default settings. The filepath to where wave files are stored is defined by `wav.at` argument. Dimensions of sound window are defined by `tlim` and `flim` and the desired overlap between windows by `ovlp=80`. In addition, time coordinates are magnified 10 times to assure that time and frequency coordinates are on the same scale.

```
# SoundShape analysis
SS.interSp<-SoundShape(mag.time=10,tlim=c(0,0.5),flim=c(0,5),
ovlp=80,plot.exp=F, wav.at="C:/R/Physalaemus - interspecific")
```

By assigning the result of `SoundShape` to an R object, it is now possible to visualize the complete set of unaligned semilandmarks before running the Generalized Procrustes Analysis (GPA) (see Figures 35 and 36). However there is no need to create these plots at this point, so we move on to the GPA analysis and then visualize the aligned semilandmarks along with consensus shape.

As discussed in the previous section, GPA of semilandmark coordinates from SoundShape yields better results when SMs are allowed to slide along the tangent of the curve of relative amplitude. We therefore proceed with the same approach by using a minimum Procrustes distance criterion to slide semilandmarks 2 to 8, and 10 to 16 (see Figures 13 to 16 for SM numbers).

R code: Sliding semilandmarks are defined by the function `define.sliders` and stored in the matrix `SS.sliders`. GPA analysis is conducted by the function `gpagen` and minimum Procrustes distance is chosen by calling the argument `ProcD = TRUE`. Aligned SMs are visualized using the matrices `$coords` and `$consensus` stored in the output of `gpagen`.

```
# SoundShape - Perform Generalized Procrustes Analysis on
semilandmarks

# Define sliders
sliders.SS=rbind(define.sliders(1:9),define.sliders(c(9:16,1)))

# SoundShape - GPA with semilandmarks allowed to slide using
Procrustes distance criterion
SS.gpa.interSp <- gpagen(SS.interSp, PrinAxes = FALSE,
print.progress = FALSE, curves = sliders.SS, ProcD = TRUE)

# SoundShape - Plot new semilandmark configurations and
consensus shape
plot(SS.gpa.interSp$coords[,1,], SS.gpa.interSp$coords[,2,],
pch=21, cex=0.9, xlab="Time coordinates",
ylab="Frequency coordinates",
main="Semilandmarks after GPA + sliding")
points(SS.gpa.interSp$consensus, pch=21, bg="orange", cex=1.1)
```

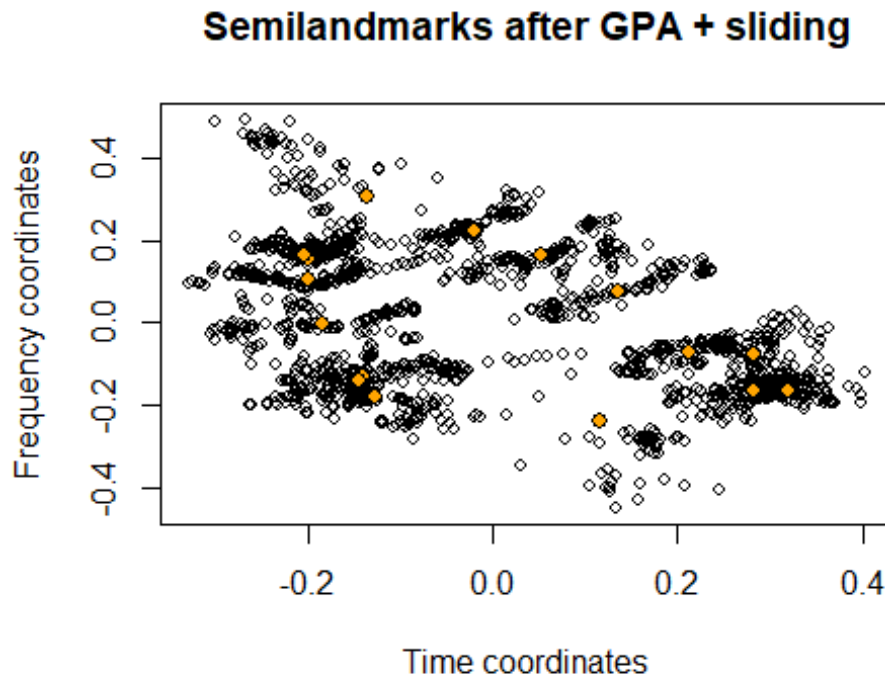


Figure 49: Aligned semilandmarks acquired by SoundShape function using advertisement calls of eight *Physalaemus* species. Hollow circles represent complete set of SMs after Generalized Procrustes Analysis using minimum Procrustes distance. Orange dots represent consensus shape calculated from Procrustes coordinates.

Once the GPA is concluded, we can proceed with the Principal Components Analysis (PCA) of the Procrustes residuals matrix (*i.e.* Procrustes coordinates). The PCA output is used to generate broken stick models (Figure 50) in order to retain only nontrivial and interpretable Principal Components (PCs).

R code: Principal Components Analysis `prcomp` is conducted on Procrustes coordinates resulting from GPA analysis. PCA output is assigned to a new object, which is used to calculate a broken stick model (Figure 50).

```
# SoundShape - Principal Components Analysis (PCA) using
# Procrustes coordinates
SS.pca.interSp <- prcomp(two.d.array(SS.gpa.interSp$coords))

# SoundShape - Broken stick model for relative variances in
# each PCs
bsm(SS.pca.interSp,main="Broken stick model - PCA SoundShape")
```

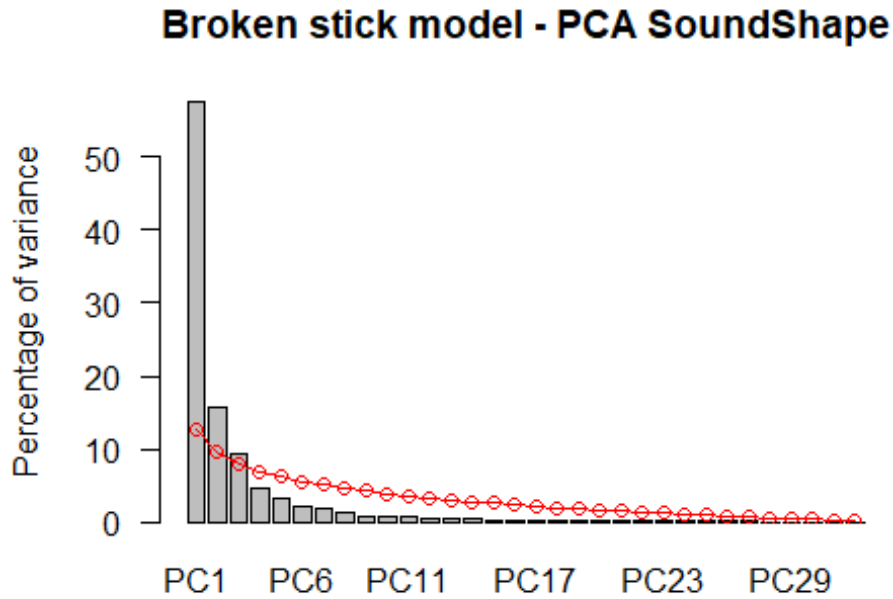


Figure 50: Eigenvalues as percentage of variance (grey bars) and broken stick model (red line and dots) computed from PCA using Procrustes coordinates. Data acquired by `SoundShape` function using the advertisement call of eight *Physalaemus* species and 2D residuals after GPA + sliding.

The visual inspection of broken stick models of variance from the PCA of `SoundShape` coordinates (Figure 50) allow us to consider PC1, PC2 and PC3 as non trivial and interpretable, in which PC1 represent 57.6% variance, PC2 represent 15.7% variance and PC3 represent 9.3% variance. However, it is not the aim of this section to extensively explore the results of PCA analysis. Therefore, we proceed with the ordination plot of PC1 and PC2 only, in which both axis encompass 73.3% of the whole variance in our dataset (Figure 51). Deformation grids (Figure 52) relative to minimum and maximum scores of Principal Components 1 and 2 are presented after the ordination plots.

Note: Although we only explore the ordination of PC1 and PC2 scores for illustrative purposes, it is strongly recommended that researchers present plots on all PCs retained by the broken stick model.

R code: Given that this empirical study encompass a slightly larger dataset, it is worth creating a loop to acquire a `data.frame` of groups from the list of files (`list.files`) being analyzed. The groups created are used as arguments by the `pca.plot` function, in which species and convex hulls are represented by the same colors, and points represent

notes analyzed by SoundShape function. The legend was created outside R platform and is included as a separate file.

```
# Create groups to use
filelist <- list.files("C:/R/Physalaemus - interspecific",
pattern=".wav")

for(i in seq_along(filelist)){
  if(!exists("File_name")){File_name<-vector(mode="character",
length=length(filelist))}
  if(!exists("Ind")){Ind <- vector(mode="character",
length=length(filelist))}
  if(!exists("Loc")){Loc <- vector(mode="character",
length=length(filelist))}
  if(!exists("Sp")){Sp <- vector(mode="character",
length=length(filelist))}

  File_name[i] <- filelist[i]
  ID.split <- strsplit(File_name[i], '[_]')[[1]]
  Ind[i] <- ID.split[2]
  Loc[i] <- substr(ID.split[2], 1, nchar(ID.split[2])-1)
  Sp[i] <- sub("P-", "P. ", ID.split[1])

  rm(ID.split)

  interSp.gr <- data.frame(Sp, Ind, Loc, File_name)
}
rm(File_name, Ind, Loc, Sp) # remove objects created to
prevent errors

# Load legend as image file
leg.interSp.orig <- imager::load.image("C:/R/Physalaemus -
interspecific/legend-Physa-interSp.jpg")
leg.interSp <- imager::crop.borders(leg.interSp.orig, nx=80,
ny=70)

# SoundShape - Plot ordination of Principal Components
layout(matrix(c(1,1,2), 1, 3))
par(mgp=c(3,0.7,0))
```

```
pca.plot(PCA.out=SS.pca.interSp, groups=interSp.gr$Sp,
conv.hulls = interSp.gr$Sp, main="SoundShape PCA - Physalaemus
species", PCs=c(1,2))
plot(leg.interSp, axes=F)
```

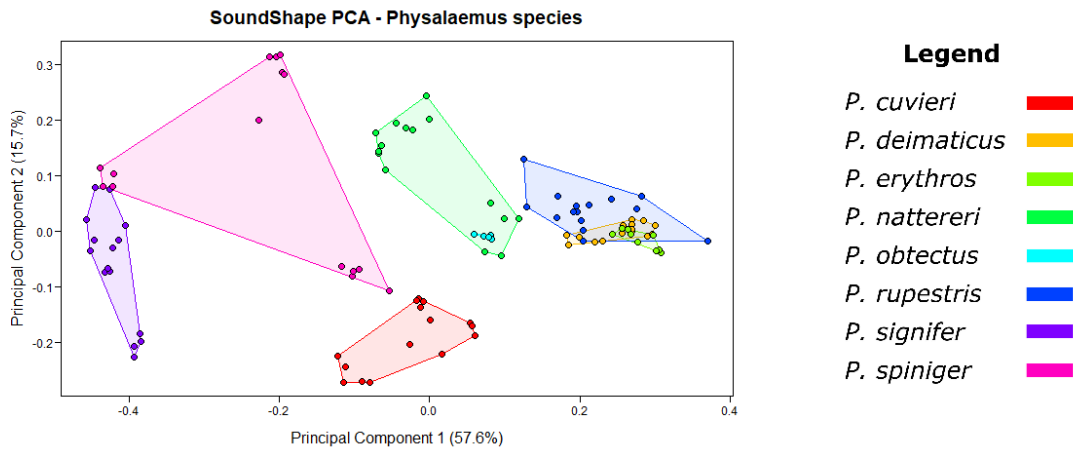


Figure 51: Individual projection of eight *Physalaemus* species onto PC1 and PC2 using Procrustes coordinates acquired by SoundShape function. Each point represent one sampled note and convex hulls comprise all notes analyzed for each species. Legend represent colors used for each species.

R code: In order to generate deformation grids (thin plates splines) for each Principal Component of PCA, we first calculate hypothetical semilandmark configurations from the Procrustes coordinates using `PCshapes` function. Deformation grids are created in relation to consensus shape of Procrustes coordinates using `plotRefToTarget` function.

```
# SoundShape - Acquire hypothetical semilandmark configurations
from Procrustes coordinates
SS.shapes.interSp <- PCshapes(SS.gpa.interSp)

# SoundShape - Principal Component 1
par(mar=c(1, 1, 1, 1), mfrow=c(2,2))
plotRefToTarget(SS.gpa.interSp$consensus,
SS.shapes.interSp$PC1min)
title(main="PC1 - Minimum")
plotRefToTarget(SS.gpa.interSp$consensus,
SS.shapes.interSp$PC1max)
title(main="PC1 - Maximum")
plotRefToTarget(SS.gpa.interSp$consensus,
SS.shapes.interSp$PC2min)
title(main="PC2 - Minimum")
plotRefToTarget(SS.gpa.interSp$consensus,
```

```
SS.shapes.interSp$PC2max)
title(main="PC2 - Maximum")
```

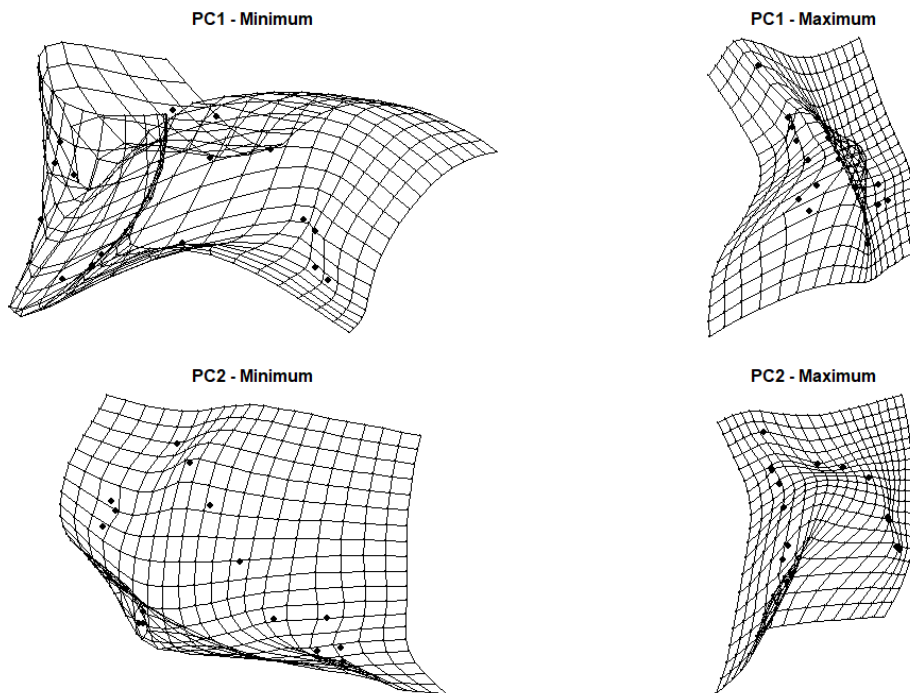


Figure 52: Semilandmarks (dots) and deformation grids in relation to consensus configuration (i.e. PC score = 0). Grids calculated using Procrustes coordinates acquired by *SoundShape* function. Data from the advertisement call of eight *Physalaemus* species. Upper panel: Minimum (left) and maximum (right) deformations from PC1 (57.6% of variation). Lower panel: Minimum (left) and maximum (right) deformation from PC2 (15.7% of variation).

The first two Principal Components from *SoundShape* analysis encompassed 73.3% of the whole variation in our dataset and the ordination of PCs yielded clear structure among species of *Physalaemus* (Figure 51). Moreover, the ordination plot show that part of this empirical data is obviously different in shape, which is somewhat expected and serve as evidence that *SoundShape* is a good method for measuring differences between fairly distinct calls. On the other hand, deformation grids (Figure 52) demonstrate such dramatic sound shape changes along the PC axis that the biological interpretation of it is somewhat complicated. By analyzing both outputs together though, the maximum deformation of PC1 seems to related to the broad band, short duration notes of *Physalaemus deimaticus* group of species. Those species overlapped entirely in the ordination of PC1 and PC2 (although PC3 is not displayed here, it provided similar structuring), which indicate that the *P. deimaticus* group share a common two dimensional sound shape.

In addition, the pair of *Physalaemus nattereri* and *P. obtectus*, also selected due to the similarity in their calls (Figure 48), also scored similar results in the PCA.

Therefore, `SoundShape` analysis might not be an accurate method when one wishes to discriminate between species with similar spectrogram outputs. However that was not true for the pair of *P. signifer* and *P. spiniger*, which also display similar spectrogram outputs, but distinct scores in the ordination plot. *P. cuvieri* had no overlap with the other *Physalaemus* species, which was expected and thus corroborate that its sound shape is unique in our sample.

Although `SoundShape` data did not result in PCA with different scores for calls with similar spectrogram outputs, it did provide a nice structuring of shapes according to the sonic structure of the analyzed calls, especially along the PC1 axis. Species plotted on the high PC1 scores were the ones with short pulsed notes (*i.e.* *Physalaemus deimaticus* group), or short harmonically structured notes (*i.e.* *P. nattereri* and *P. obtectus*). Species with medium duration and clear harmonic structure scored low PC1 values, which placed them on the left side of the ordination plot. In the light of this, the dramatic deformation observed in the thin plate splines from PC1 scores (Figure 52) is likely related to the presence of harmonic structure of notes.

Overall, `SoundShape` analysis resulted in ordination plots with high statistical power, thus encompassing the majority of variation embedded in our dataset (*i.e.* 73.3% variation). In addition, the two dimensional approach was successful in the opposite direction as originally intended, in the sense that it served greatly in structuring species with similar call shapes, in addition to scoring different PC values according to the sonic structure of calls (*i.e.* presence of harmonic structures). Although the implications of this method are yet to be revealed, `SoundShape` approach seems a straightforward analysis that simplify spectrogram variation into two dimensional coordinates and serve well the purpose of structuring species with similar calls.

eigensound analysis – In order to proceed with the `eigensound` analysis, we must first define the sampling grid to be applied over the three dimensional sound surface. Given the medium duration of notes selected for this study, we managed to employ a rather small sound window (*i.e.* time X-axis from 0 to 0.5 s; frequency Y-axis from 0 to 5 kHz; Figure 48), which will require a customized sampling grid smaller than `eigensound` default.

The sampling grid is defined using the time `$time` and frequency `$freq` coordinates embedded in a `spectro` object created from a sample wave file. Additionally, we can also employ the complete set of semilandmarks acquired by

SoundShape function in order to better define the number of cells on each axis of the sampling grid and choose between a linear and a logarithmic scale on the time X-axis.

R code: A sample note from the advertisement call of *Physalaemus deimaticus* is read with `readWave` function and the spectrogram object is generated by `spectro` function. New sequences `seq` are created and used to subset the original time and frequency coordinates, thus simplifying the grid of amplitude values. The new sampling grid is then plotted along with the complete set of semilandmarks from SoundShape analysis.

```
# Read sample wave to define number of cells on each side
sampleWave <- readWave("C:/R/Physalaemus - interspecific/P-cuvieri_C-Chagas1_005_CBUFG-142.wav")

# Create spectro object from sample wave
spec.sample <- spectro(sampleWave, plot=FALSE, tlim=c(0,0.5),
flim=c(0,5), f=44100, wl=512, ovlp=80)

# Dimensions of original grid of amplitude coordinates
dim(spec.sample$amp)

## [1] 59 211

# New sequences to use as frequency coordinates
freq.seq <- seq(1, length(spec.sample$freq), length = 30)
time.seq <- 10^(seq(log10(1), log10(length(spec.sample$time)),
length.out = 100)) #log sequence

freq.sub <- spec.sample$freq[freq.seq]
time.sub <- spec.sample$time[time.seq]

# Visualize complete set of semilandmarks to define between
log and linear X-axis
par(mfrow=c(1,2))
plot((SS.interSp[, 1, ]), SS.interSp[, 2, ], pch=21, cex=0.9,
ylim=c(0,5), xlim=c(0,5), xlab="Time coordinates",
ylab="Frequency coordinates",
main="Semilandmarks from SoundShape")
points(mshape(SS.interSp), pch=21, bg="orange", cex=1.1)
```

```
# Visualize sampling grid of semilandmark coordinates
plot(x=c(0,5), y=c(0,5), type='n', las=1,
xlab="Time coordinates", ylab="Frequency coordinates",
main="Simplified grid of semilandmark coordinates")
abline(v=time.sub*10, h=freq.sub, lwd=0.1)
```

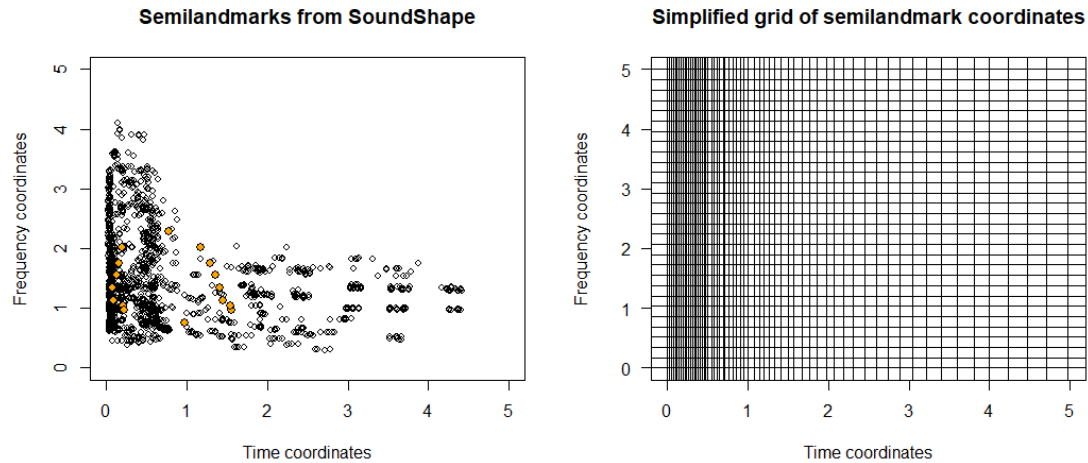


Figure 53: Left: Complete set of semilandmarks (hollow circles) acquired by `SoundShape` function from advertisement calls of eight *Physalaemus* species. Orange dots represent mean configuration calculated from the complete set of semilandmarks. Right: Simplified grid of semilandmark coordinates using a logarithmic sequence on the time X-axis. Grid encompass 100 cells on the time X-axis and 30 cells on the frequency Y-axis, enabling the acquisition of 3000 semilandmarks from the sound surface. **Note:** Time coordinates have been magnified 10 times so that time and frequency coordinates will be on the same scale.

The disposition of `SoundShape` semilandmarks along the sound window (Figure 53) demonstrate that the majority of SMs is concentrated between 0 and 0.1 s (note that time coordinates have been magnified 10 times, thus coordinates on time X-axis range from 0 to 5 instead of the true range of sound windows between 0 and 0.5 s). Therefore, we opted for a logarithmic scale on the time X-axis, which emphasize the predominance of short duration notes rather than the few longer ones. The number of cells on each axis was set with 100 cells on the time X-axis and 30 cells on the frequency Y-axis, assuring the appropriate sampling of time coordinates before 0.1 s. This simplified grid enabled the acquisition of 3000 variables (*i.e.* semilandmarks – SMs) from the three dimensional surface of sound. Given that the original grid of amplitude values is represented by 59 rows (frequency Y-axis), and 211 columns (time X-axis), our simplified grid represent a 75% reduction in data dimensionality (12449 variables in the original grid).

R code: `eigensound` function require stating the filepath `wav.at` from which wave files will be analyzed. In addition, we also define the dimensions intended for the sound window (`tlim` and `flim`) and the time resolution through overlapping windows (`ovlp=80`). Number of cells on the X and Y-axis are defined by `x.lenght` and

y.lenght, respectively. A logarithmic scale on the X-axis is chosen by calling `log.scale=TRUE`.

```
# eigensound analysis
eig.interSp <- eigensound(tlim=c(0,0.5), flim=c(0,5), ovlp=80,
x.length=100, y.length=30, log.scale=TRUE, plot.exp = FALSE,
wav.at="C:/R/Physalaemus - interspecific")
```

At this point, it is also possible to visualize the complete set of unaligned semilandmarks and the mean shape before running the Generalized Procrustes Analysis (GPA) (see Figure 36). However there is no need to create these plots at this point, so we move on to the GPA analysis and then visualize the aligned semilandmarks along with consensus shape (Figure 54).

R code: Aligned SMs are visualized using the matrices `$coords` and `$consensus` stored in the output of `gpagen`.

```
# eigensound - Perform Generalized Procrustes Analysis on
semilandmarks
eig.gpa.interSp <- gpagen(eig.interSp, PrinAxes = FALSE,
print.progress = FALSE)

# eigensound - Plot new semilandmark configurations and
consensus shape
par(mfrow=c(1,2))
scatter3D(x=eig.gpa.interSp$coords[,1,],
y=eig.gpa.interSp$coords[,2,], z=eig.gpa.interSp$coords[,3,],
theta=30, phi=35, resfac=1, r=3, expand=0.5, scale=T, axes=T,
cex.axis=0.7, ticktype="detailed", nticks=4, pch=21,
col=spectro.colors(n=100), cex=0.5, xlab="Time (s)",
ylab="Frequency (kHz)", zlab="Amplitude (dB)",
clab=expression('Amplitude dB'),
main=expression('Semilandmarks after GPA'))
scatter3D(x=eig.gpa.interSp$consensus[,1,],
y=eig.gpa.interSp$consensus[,2,],z=eig.gpa.interSp$consensus[,3,],
theta=30, phi=35, resfac=1, r=3, expand=0.5, scale=T, axes=T,
cex.axis=0.7, ticktype="detailed", nticks=4, pch=21,
col=spectro.colors(n=100), cex=0.5, xlab="Time (s)",
ylab="Frequency (kHz)", zlab="Amplitude (dB)",
```

```
clab=expression('Amplitude dB'),
main=expression('Consensus shape'))
```

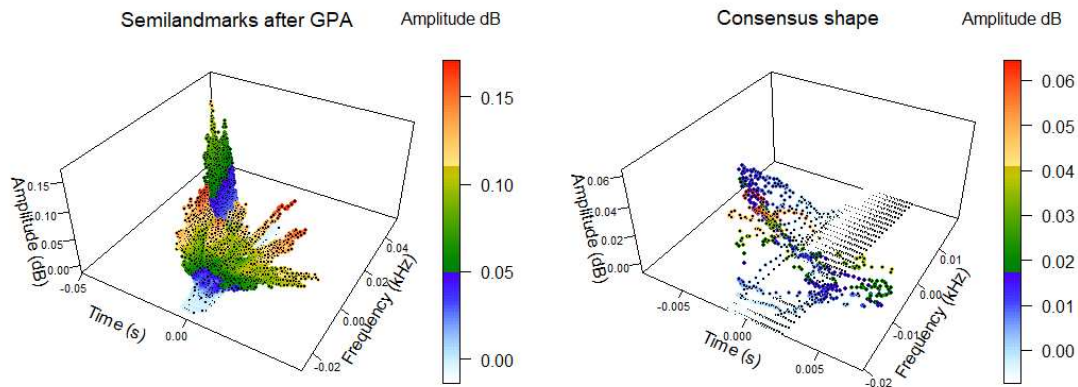


Figure 54: Semilandmarks acquired by eigensound function using advertisement calls of eight *Physalaemus* species. Left: Complete set of three dimensional semilandmarks after Generalized Procrustes Analysis. Right: Consensus shape calculated from the aligned set of three dimensional semilandmarks.

Once the GPA is concluded, we can proceed with the Principal Components Analysis (PCA) of the Procrustes residuals matrix (*i.e.* Procrustes coordinates). The PCA output is used to generate broken stick models (Figure 55) in order to retain only nontrivial and interpretable Principal Components (PCs).

R code: Principal Components Analysis `prcomp` is conducted on Procrustes coordinates resulting from GPA analysis. PCA output is assigned to a new object, which is used to calculate a broken stick model (Figure 55).

```
# eigensound - Principal Components Analysis (PCA) using
# Procrustes coordinates
eig.pca.interSp <- prcomp(two.d.array(eig.gpa.interSp$coords))

# eigensound - Broken stick model for relative variances in ea
# ch PCs
bsm(eig.pca.interSp, main="Broken stick model - eigensound after
GPA")
```

Broken stick model - eigensound after GPA

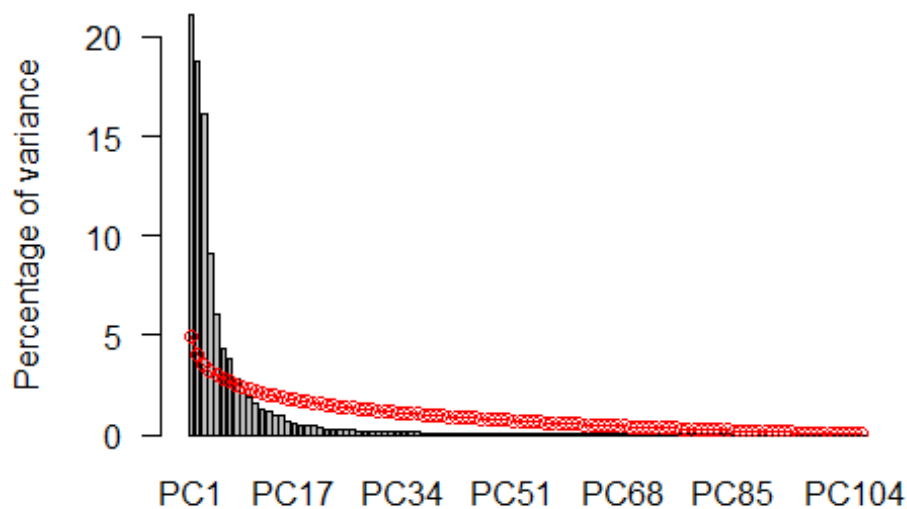


Figure 55: Eigenvalues as percentage of variance (grey bars) and broken stick model (red line and dots) computed from PCA using Procrustes coordinates. Data acquired by `eigensound` function using the advertisement call of eight *Physalaemus* species.

Although our broken stick model (Figure 55) retained the first eight Principal Components as nontrivial and interpretable, the PC6, PC7 and PC8 represent less than 5% variance each. In addition, only PC1, PC2 and PC3 represent more than 15% variance each, therefore these are the recommended Principal Components to use in the ordination plot. However, it is not the aim of this section to extensively explore the results of any PCA analysis. Therefore, we proceed with the ordination plot of PC1 and PC2 only, in which both axis encompass 39% of the whole variance in our dataset (Figure 56). Hypothetical semilandmark configurations (Figure 57) relative to Principal Components 1 and 2 will follow the ordination plots.

R code: The `data.frame` of groups created in previous code chunks are used as arguments by the `pca.plot` function. Species and convex hulls are represented by the same colors. Points represent notes analyzed by `eigensound` function. The legend was created outside R platform and is displayed from a separate file.

```
# eigensound - PCA Plot (ordination of Principal Components)
layout(matrix(c(1,1,2), 1, 3))
par(mgp=c(3,0.7,0))
pca.plot(PCA.out=eig.pca.interSp, groups=interSp.gr$Sp,
```

```
conv.hulls = interSp.gr$Sp, main="eigensound PCA - Physalaemus
species")
plot(leg.interSp, axes=F)
```

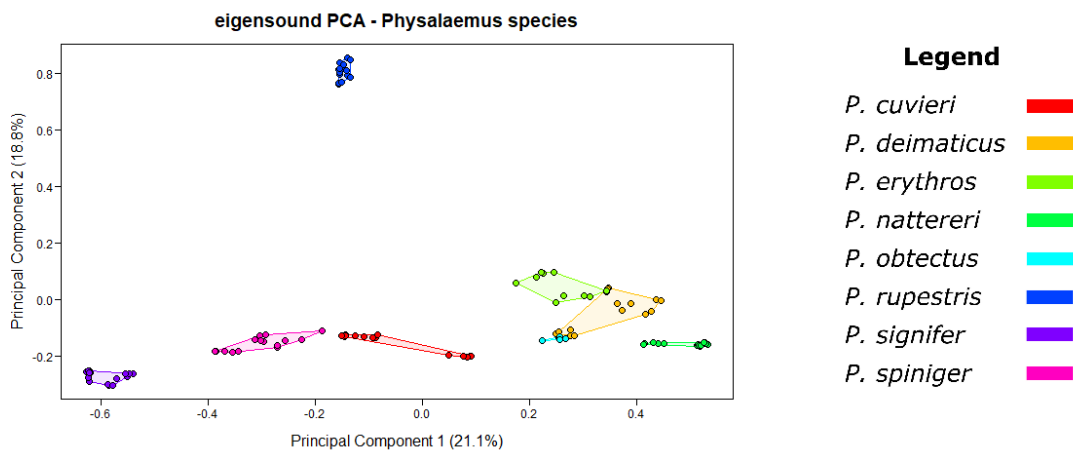


Figure 56: Individual projection of eight *Physalaemus* species onto PC1 and PC2 using Procrustes coordinates acquired by `eigensound` function. Each point represent one sampled note and convex hulls comprise all notes analyzed for each species. Legend represent colors used for each species.

R code: Hypothetical semilandmark configurations for each Principal Component of PCA are calculated from Procrustes coordinates by the function `PCshapes` and plotted with `scatter3D` function.

```
# Acquire hypothetical semilandmark configurations from
Procrustes coordinates
eig.shapes.interSp <- PCshapes(eig.gpa.interSp)

# Hypothetical three dimensional semilandmark configurations
from PCs of eigensound analysis
par(mar=c(1, 1, 1, 1), mfrow=c(2,2))
scatter3D(x=eig.shapes.interSp$PC1min[,1],
y=eig.shapes.interSp$PC1min[,2],z=eig.shapes.interSp$PC1min[,3],
theta=30, phi=35, resfac=1, r=3, expand=0.5, scale=T, axes=T,
cex.axis=0.7, ticktype="detailed", nticks=4, pch=21,
col=spectro.colors(n=100), cex=0.5, xlab="x", ylab="y",zlab="N",
clab=expression('Amplitude'), main=expression('PC1 - Minimum'))

scatter3D(x=eig.shapes.interSp$PC1max[,1],
y=eig.shapes.interSp$PC1max[,2],z=eig.shapes.interSp$PC1max[,3],
theta=30, phi=35, resfac=1, r=3, expand=0.5, scale=T, axes=T,
cex.axis=0.7, ticktype="detailed", nticks=4, pch=21,
col=spectro.colors(n=100), cex=0.5,xlab="x", ylab="y", zlab="N",
```

```

clab=expression('Amplitude'), main=expression('PC1 - Maximum'))

scatter3D(x=eig.shapes.interSp$PC2min[,1],
y=eig.shapes.interSp$PC2min[,2],z=eig.shapes.interSp$PC2min[,3],
theta=30, phi=35, resfac=1, r=3, expand=0.5, scale=T, axes=T,
cex.axis=0.7, ticktype="detailed", nticks=4, pch=21,
col=spectro.colors(n=100),cex=0.5, xlab="x", ylab="y", zlab="N",
clab=expression('Amplitude'), main=expression('PC2 - Minimum'))

scatter3D(x=eig.shapes.interSp$PC2max[,1],
y=eig.shapes.interSp$PC2max[,2],z=eig.shapes.interSp$PC2max[,3],
theta=30, phi=35, resfac=1, r=3, expand=0.5, scale=T, axes=T,
cex.axis=0.7, ticktype="detailed", nticks=4, pch=21,
col=spectro.colors(n=100),cex=0.5, xlab="x", ylab="y", zlab="N",
clab=expression('Amplitude'), main=expression('PC2 - Maximum'))

```

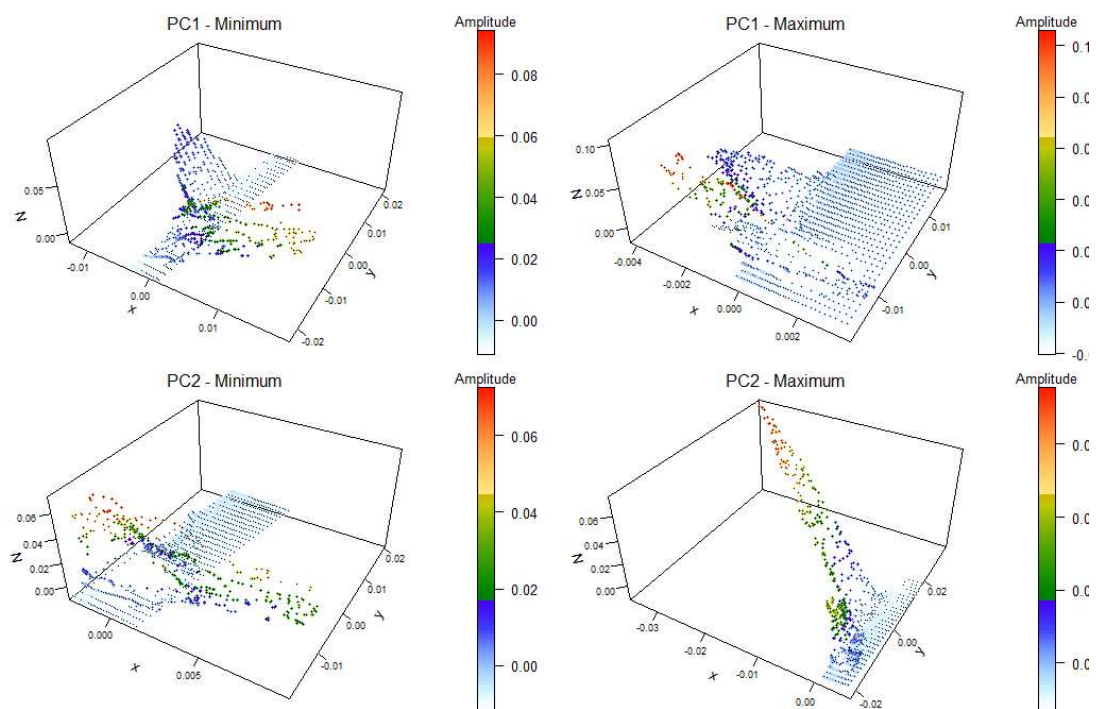


Figure 57: Hypothetical semilandmark configurations from minimum (left) and maximum (right) scores of PCA using Procrustes coordinates acquired by `eigensound` function. Data from the advertisement call of eight *Physalaemus* species. Upper panel: Hypothetical semilandmark configurations from PC1 (21.1% variance). Lower panel: Minimum (left) and maximum (right) semilandmark configurations from PC2 (18.7% variance).

Ordination of Principal Components from `eigensound` analysis resulted in remarkable structuring of individuals belonging to the same species (Figure 56). Aside from resulting in different scores for species with distinct calls (e.g. *Physalaemus rupestris* and *P. cuvieri*), the `eigensound` method was also sensitive to species with

similar advertisement calls (e.g. *P. deimaticus* and *P. erythros*; *P. nattereri* and *P. obtectus*). In addition, the three dimensional point coordinates scored distinct values according to the acoustic structure of each species call. Those with harmonically structured calls scored low PC2 values, thus structuring them at the bottom of the ordination plot (i.e. from left to right: *Physalaemus signifer*, *P. spiniger*, *P. cuvieri*, *P. obtectus* and *P. nattereri*). This is in agreement with the results from SoundShape and the apparent harmonic structure displayed in the hypothetical semilandmark configuration of minimum PC2 scores (Figure 57).

Despite the semilandmark configurations from PC1 scores being non intuitive, the placement of species along the PC1 axis indicate that this Component is related to the note duration among species (Figure 56). *Physalaemus obtectus* and *P. nattereri*, for instance, have very similar advertisement calls characterized by short notes with harmonic structures. That granted these species high PC1 scores and low PC2 scores, which places them at the bottom right hand side of the ordination plot. *Physalaemus spiniger* and *P. signifer*, on the other hand, both have harmonically structured calls with medium duration, which granted them low PC1 and PC2 scores and structured these species at the lower left hand side of the ordination plot.

Important to note that the two dimensional sound shape of species with similar calls (i.e. *Physalaemus deimaticus* group; *P. nattereri* and *P. obtectus*) resulted in similar PC scores, which structured these species together in the SoundShape data (Figure 51). The eigensound analysis, in turn, resulted in distinct PC scores for every species employed in this empirical study, thus indicating that the three dimensional sound surface is likely species specific within the genus *Physalaemus*. In addition, *P. rupestris* presented a unique sound surface, which granted this species the highest PC2 scores and a distinct placement on the plot. However, the interpretation of hypothetical semilandmark configurations relative to the maximum PC2 score is not straightforward. Yet the inspection of jpeg images generated by SoundShape yields a reasonable comparison between the sound shapes (Figure 48), in which is possible to note that *P. rupestris* is the only *Physalaemus* species in our intraspecific study that has clear pulsed structure in the spectrogram output of its call. Therefore, our interspecific study on call shape variation revealed that the eigensound method is sensitive to both the sonic structure and the specificity of advertisement calls in the *Physalaemus* species studied.

Interpopulational variation in the sound shape of *Physalaemus deimaticus* Sazima & Caramaschi 1986

In order to evaluate how `SoundShape` and `eigensound` functions perform in the assessment of sound shape changes between populations, we selected advertisement calls of *Physalaemus deimaticus* Sazima & Caramaschi 1986 (Anura: Leptodactylidae) from three localities. Three individuals were recorded at each locality, including the species type locality (Serra do Cipó, Morro do Pilar Municipality, Minas Gerais State, Brazil), Diamantina Municipality (Minas Gerais State, Brazil) and Pico do Itambé (Santo Antônio do Itambé Municipality, Minas Gerais State, Brazil) (see Appendix 1 from Rocha & Romano *in prep* for details on recordings). Each individual had five stereotyped notes selected according to optimal signal to noise ratio and no overlap with background individuals, thus encompassing 45 notes stored as separate `wave` files. Calls were placed at time = 0 s and the remaining sound window completed with silence. Sound windows were set to range from 0 to 0.15 seconds and from 0 to 5 kHz, which encompassed all notes from the sample. Spectrogram parameters were set with window length `wl=512`, sampling rate `f=44100` and 80% overlap `ovlp=80` (Figure 58).

All 45 notes selected were stored in a separate folder from which `SoundShape` and `eigensound` analysis will be conducted. `SoundShape` is ran under its default settings, which yields the acquisition of 16 semilandmarks (SM) from the curve of relative amplitude at -25 dB. However we must first define the sampling grid to be applied over the sound surface for the *eigesound* approach. Given the short notes selected for this study, we managed to employ a rather small sound window (*i.e.* time X-axis from 0 to 0.15 s; frequency Y-axis from 0 to 5 kHz, Figure 58), which demands a small sampling grid.

R code: A sample note from the advertisement call of *Physalaemus deimaticus* is read with `readWave` function and spectrogram plot generated using `spectro` function.

```
# Read sample wave
cipol <- readWave("C:/R/P-deimaticus - interpop/P-deimaticus_
Cipol_084_CBUFG-0173.wav")

# Spectrogram and oscillogram of a representative note of the
advertisement call of P. deimaticus
spec.cipol <- spectro(cipol, osc=T, grid=F, scale=T, f=44100,
```

```
tlim=c(0,0.15), flim=c(0,5), wl=512, ovlp=80,
main=expression(' '*italic('Physalaemus deimaticus')* ' call'))
```

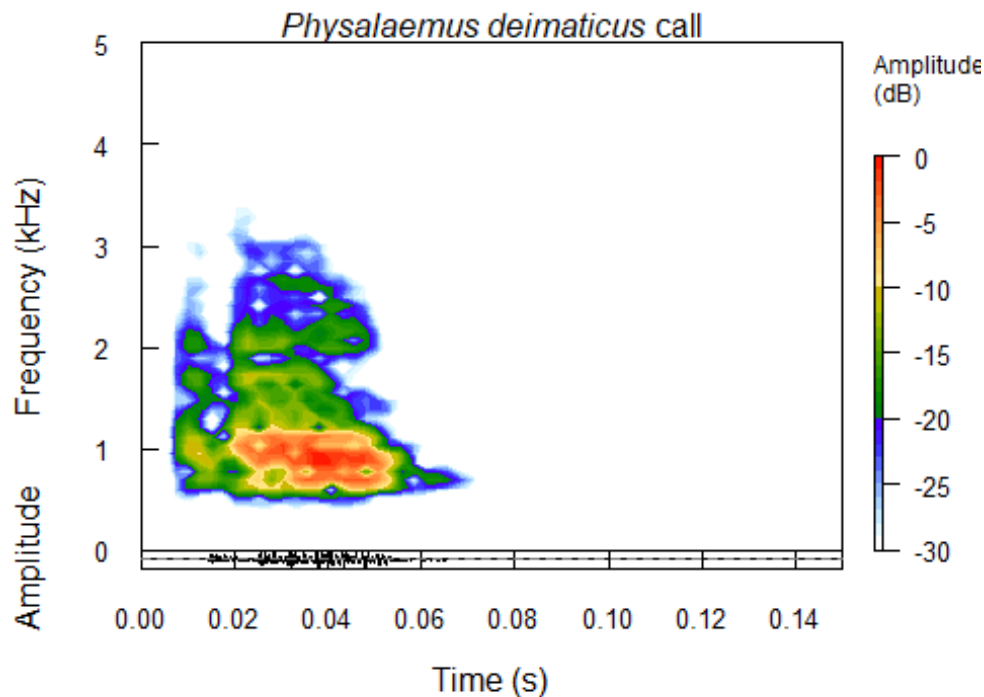


Figure 58: Spectrogram (above) and oscillogram (bellow) of a representative note from the advertisement call of *Physalaemus deimaticus* recorded at its type locality, Serra do Cipó, Morro do Pilar Municipality, Minas Gerais State, Brazil.

R code: We can verify the dimensions `dim` of the matrix of amplitude values in order to define the new sampling grid to be applied over the sound surface.

```
# Verify dimensions of matrix of amplitude values
dim(spec.cipol$amp)

## [1] 59 60
```

The original grid of amplitude values result in a dataset composed of 59 rows (frequency Y-axis) and 60 columns (time X-axis), representing a total of 3540 variables. Therefore, we run `eigensound` analysis with a simplified sampling grid of 30 cells on the frequency Y-axis, and 40 cells on the time X-axis, which result in a dataset composed of 1200 variables (*i.e.* semilandmarks – SMs), a 66.1% reduction in data dimensionality. In addition, since all notes analyzed present short duration (*i.e.* less than 0.11 s), we employ a linear scale on the time X-axis by calling the argument `log.scale=FALSE` on `eigensound` function.

Note: At this point, it is recommended to plot both the original grid of amplitude coordinates and the simplified grid to be used in `eigensound` analysis. Although this

can easily be done using the `$freq` and `$time` coordinates embedded in the `spectro` object, we will not provide this plot as it has been done in previous code chunks (see Figures 28 and 30).

R code: Both `SoundShape` and `eigensound` functions require stating the filepath `wav.at` from which wave files will be analyzed. In addition, we also define the dimensions intended for the sound window (`tlim` and `flim`) and the time resolution through overlapping windows (`ovlp=80`). Given the very small sound window, we magnify the coordinates on the time X-axis by calling `mag.time=100`, thus assuring that time and frequency coordinates are on the same scale. For more details on functionalities of `SoundShape` and `eigensound` functions (e.g. exporting jpeg images), please see Tables 2 and 3.

```
# Analysis of sound shape from three localities of Physalaemus
deimaticus

# SoundShape
SS.deim <- SoundShape(wav.at = "C:/R/P-deimaticus - interpop",
mag.time=100, tlim=c(0, 0.15), flim=c(0, 5), ovlp=80, plot.exp=FALSE)

# eigensound
eig.deim <- eigensound(wav.at="C:/R/P-deimaticus - interpop",
tlim=c(0, 0.15), flim=c(0, 5), ovlp=80, x.length=40, y.length=30,
log.scale=FALSE, plot.exp = FALSE)
```

Once acquired the point coordinates from *SoundShape* and *eigensound* methods, it is possible to visualize the complete set of unaligned semilandmarks and the mean shape before running the Generalized Procrustes Analysis (GPA) (see Figures 35 and 36). Although there is little additional information to be drawn from these plots, the visualization of the complete set of semilandmarks from *SoundShape* is very useful when defining the *eigensound* sampling grid (Figure 53). However this interpopulational study only deal with short duration notes, which makes for a straightforward definition of *eigensound* methods, so we move on to the GPA analysis and then provide the plots of aligned semilandmarks along with consensus shape (Figures 59 and 60).

As discussed in the previous section, GPA of semilandmark coordinates from *SoundShape* yields better results when SMs are allowed to slide along the curve of

relative amplitude by a minimum Procrustes distance criterion. We therefore proceed with the same approach by allowing the semilandmarks 2 to 8, and 10 to 16 to slide (see Figures 13 to 16 for SM numbers).

R code: Sliding semilandmarks are defined by the function `define.sliders` and stored in the matrix `SS.sliders`. GPA analysis is conducted by the function `gpagen` and minimum Procrustes distance is chosen by calling the argument `ProcD = TRUE`. Aligned SMs are visualized using the matrices `$coords` and `$consensus` stored in the output of `gpagen`.

```
# Define sliders
SS.sliders=rbind(define.sliders(1:9),define.sliders(c(9:16,1)))

# GPA with semilandmarks allowed to slide using minimum
Procrustes distance criterion
SS.gpa.deim <- gpagen(SS.deim, PrinAxes = FALSE,
print.progress = FALSE, curves = SS.sliders, ProcD = TRUE)

# Visualize aligned semilandmark configurations and consensus
shape
plot(SS.gpa.deim$coords[,1,],SS.gpa.deim$coords[,2,],pch = 21,
cex=0.9, xlab="Time coordinates", ylab="Frequency coordinates",
main="Aligned semilandmarks from SoundShape")
points(SS.gpa.deim$consensus, pch=21, bg="orange", cex=1.1)
```

Aligned semilandmarks from SoundShape

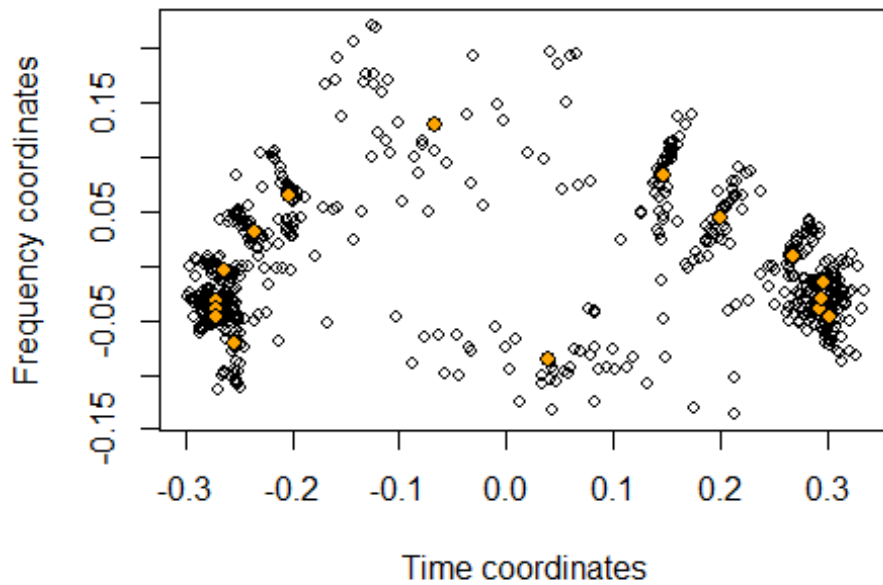


Figure 59: Semilandmarks acquired by SoundShape function using advertisement calls of *Physalaemus deimaticus* from three populations. Hollow circles represent complete set of SMs after Generalized Procrustes Analysis using minimum Procrustes distance. Orange dots represent consensus shape calculated from Procrustes coordinates.

```
# GPA with semilandmarks acquired by eigensound
eig.gpa.deim <- gpagen(eig.deim, PrinAxes = FALSE,
print.progress = FALSE)

# Visualize aligned semilandmark configurations and consensus
shape
par(mfrow=c(1,2))
scatter3D(x=eig.gpa.deim$coords[,1,],
y=eig.gpa.deim$coords[,2,],z=eig.gpa.deim$coords[,3,], theta=30,
phi=35, resfac=1, r=3, expand=0.5, scale=T, axes=T,cex.axis=0.7,
ticktype="detailed",nticks=4, pch=21, col=spectro.colors(n=100),
cex=0.5, xlab="Time coordinates",ylab="Frequency coordinates",
zlab="Amplitude coordinates", clab=expression('Amplitude'),
main=expression('Semilandmarks after GPA'))

scatter3D(x=eig.gpa.deim$consensus[,1],
y=eig.gpa.deim$consensus[,2], z=eig.gpa.deim$consensus[,3],
theta=30, phi=35, resfac=1, r=3, expand=0.5, scale=T, axes=T,
cex.axis=0.7, ticktype="detailed", nticks=4, pch=21,
col=spectro.colors(n=100), cex=0.5, xlab="Time coordinates",
```

```
ylab="Frequency coordinates",zlab="Amplitude coordinates",
clab=expression('Amplitude'),main=expression('Consensus shape'))
```

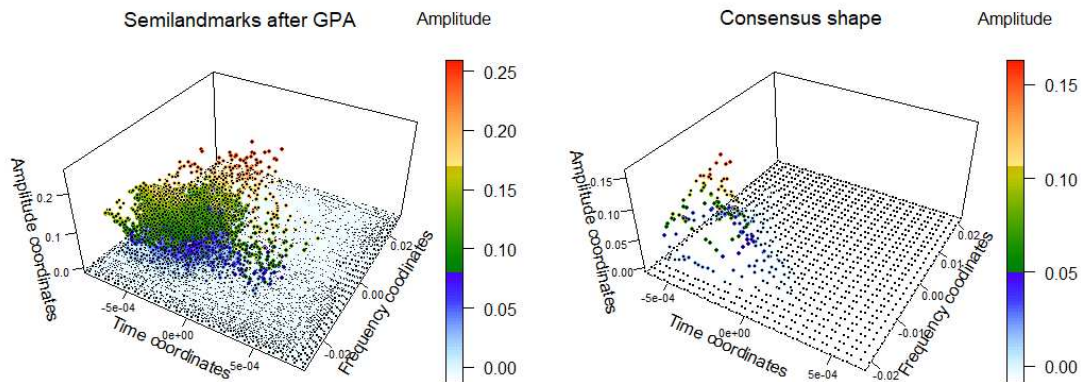


Figure 60: Semilandmarks acquired by `eigensound` function using advertisement calls of *Physalaemus deimaticus* from three populations. Left: Complete set of three dimensional semilandmarks after Generalized Procrustes Analysis. Right: Consensus shape calculated from the aligned set of three dimensional semilandmarks.

Once the GPA is concluded, we can proceed with the Principal Components Analysis (PCA) using the Procrustes coordinates. The PCA output is used to generate broken stick models (Figure 61) in order to retain only nontrivial and interpretable Principal Components (PCs).

R code: Principal Components Analysis `prcomp` is conducted on Procrustes coordinates resulting from GPA analysis. PCA output is assigned to a new object, which is used to generate broken stick models.

```
# Principal Components Analysis (PCA) using Procrustes
coordinates from SoundShape and eigensound
SS.pca.deim <- prcomp(two.d.array(SS.gpa.deim$coords))
eig.pca.deim <- prcomp(two.d.array(eig.gpa.deim$coords))

# Broken stick model for relative variances in each PCs
par(mfrow=c(1,2))
bsm(SS.pca.deim, main="SoundShape - Broken stick model")
bsm(eig.pca.deim, main="eigensound - Broken stick model")
```

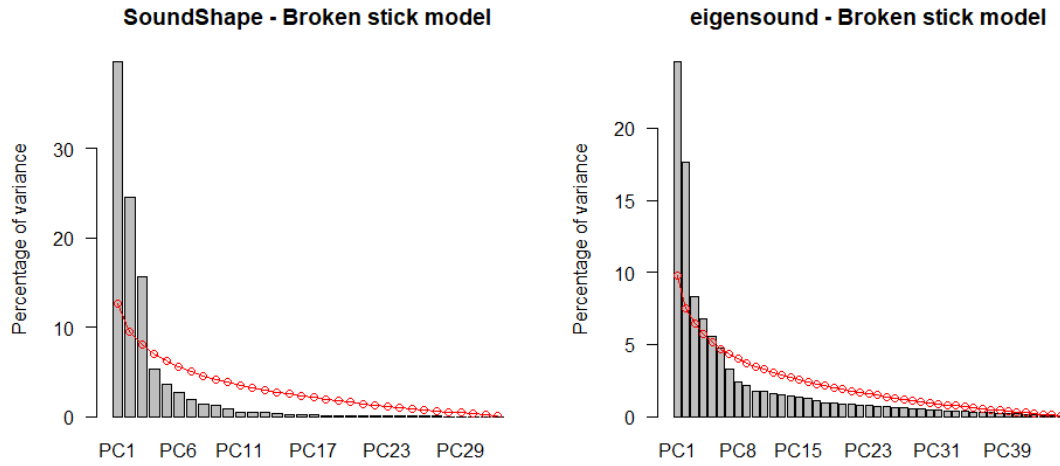


Figure 61: Eigenvalues as percentage of variance (grey bars) and broken stick model (red line and dots) computed from PCA using Procrustes coordinates. Left: Data acquired by `SoundShape` function. Right: Data acquired by `eigensound` function.

The visual inspection of broken stick models of variance (Figure 61) allow us to retain PC1, PC2 and PC3 from the PCA of `SoundShape` coordinates, in which PC1 represent 39.7% variance, PC2 represent 24.6% variance and PC3 represent 15.6% variance. As in the previous empirical study, only PC1 and PC2 are displayed as an ordination plot, in which both axis encompass 64.3% of the whole variance in our dataset (Figure 62).

Regarding the `eigensound` coordinates, although the broken stick model retained the first six Principal Components as nontrivial and interpretable, PC5 and PC6 represent less than 5% variance each. In addition, the variance expressed by those axis is approximately the same as expected from a data randomly distributed, so it is not recommended to produce ordination plots with these PCs. Thus, we proceed with the ordination plot of PC1 and PC2, in which both axis encompass 42.2% of the whole variation in our dataset (Figure 64).

Deformation grids (Figure 63) and hypothetical semilandmark configurations (Figure 65) relative to minimum and maximum scores of Principal Components 1 and 2 will follow the ordination plots for the `SoundShape` and `eigensound` analysis, respectively.

R code: Given that this empirical study encompass a slightly larger dataset, it is worth creating a loop to acquire a `data.frame` of groups from the list of files `list.files` being analyzed. The groups created are used as arguments by the `pca.plot` function, in which each individual is represented by colored points and each population is delimited

by a convex hull. The legend was created outside R platform and is included as a separate file.

```
# Create groups to use as arguments for ordination plots
filelist <- list.files("C:/R/P-deimaticus - interpop",
pattern=".wav")

for(i in seq_along(filelist)){
  if(!exists("File_name")){File_name<-vector(mode="character",
length=length(filelist))}
  if(!exists("Ind")){Ind <- vector(mode="character",
length=length(filelist))}
  if(!exists("Loc")){Loc <- vector(mode="character",
length=length(filelist))}
  if(!exists("Sp")){Sp <- vector(mode="character",
length=length(filelist))}

  File_name[i] <- filelist[i]
  ID.split <- strsplit(File_name[i], '[_]')[[1]]
  Ind[i] <- ID.split[2]
  Loc[i] <- substr(ID.split[2], 1, nchar(ID.split[2])-1)
  Sp[i] <- sub("P-", "P. ", ID.split[1])

  rm(ID.split)

  deim.gr <- data.frame(Sp, Ind, Loc, File_name)
}
rm(File_name, Ind, Loc, Sp)

# Load legend as jpeg image and store as R object
leg.interpop.orig <- imager::load.image("C:/R/P-deimaticus -
interpop/legend-Physa-interpop.jpg")
leg.interpop <- imager::crop.borders(leg.interpop.orig,nx=100,
ny=60)

# Plot ordination of Principal Components from SoundShape
layout(matrix(c(1,1,2), 1, 3))
par(mfp=c(3,0.7,0))
pca.plot(PCA.out=SS.pca.deim, groups=deim.gr$Ind,
```

```
conv.hulls = deim.gr$Loc, leg=F, main=expression('SoundShape PCA
- '*italic('P. deimaticus')* ' populations')) # Legend
```

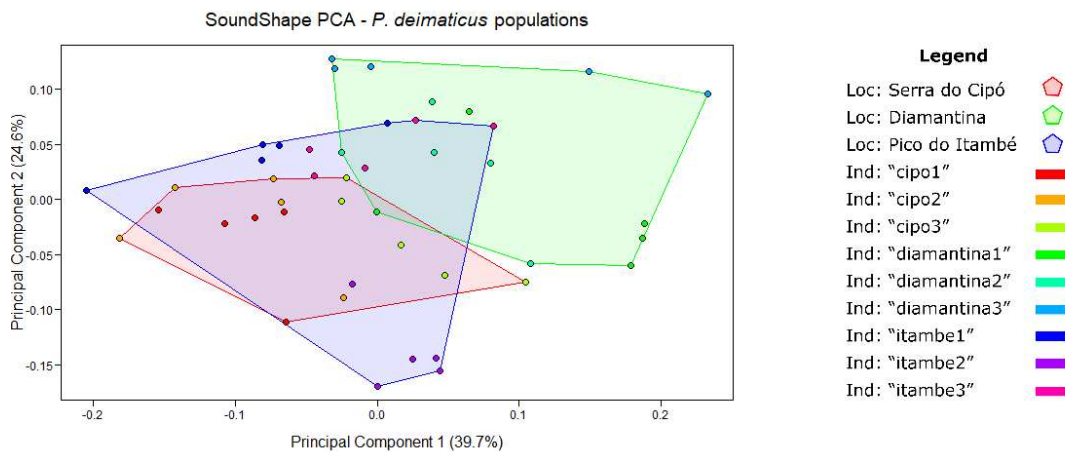


Figure 62: Individual projection of *Physalaemus deimaticus* individuals onto PC1 and PC2 axes using Procrustes coordinates acquired by SoundShape function. Each point represent one sampled note and convex hulls comprise all notes analyzed for each population. Legend represent colors used for each individual and localities encompassed by convex hulls. Ind = individual. Loc = locality.

R code: In order to generate deformation grids (thin plates splines) for each Principal Component of PCA, we first calculate hypothetical semilandmark configurations from the Procrustes coordinates. Deformation grids are generated in relation to consensus shape of Procrustes coordinates using `plotRefToTarget` function.

```
# Acquire hypothetical semilandmark configurations from
Procrustes coordinates
SS.shapes.deim <- PCshapes(SS.gpa.deim)

# Deformation grids from Principal Components of SoundShape
analysis
par(mar=c(1, 1, 1, 1), mfrow=c(2,2))
plotRefToTarget(SS.gpa.deim$consensus, SS.shapes.deim$PC1min)
title(main="PC1 - Minimum")
plotRefToTarget(SS.gpa.deim$consensus, SS.shapes.deim$PC1max)
title(main="PC1 - Maximum")
plotRefToTarget(SS.gpa.deim$consensus, SS.shapes.deim$PC2min)
title(main="PC2 - Minimum")
plotRefToTarget(SS.gpa.deim$consensus, SS.shapes.deim$PC2max)
title(main="PC2 - Maximum")
```

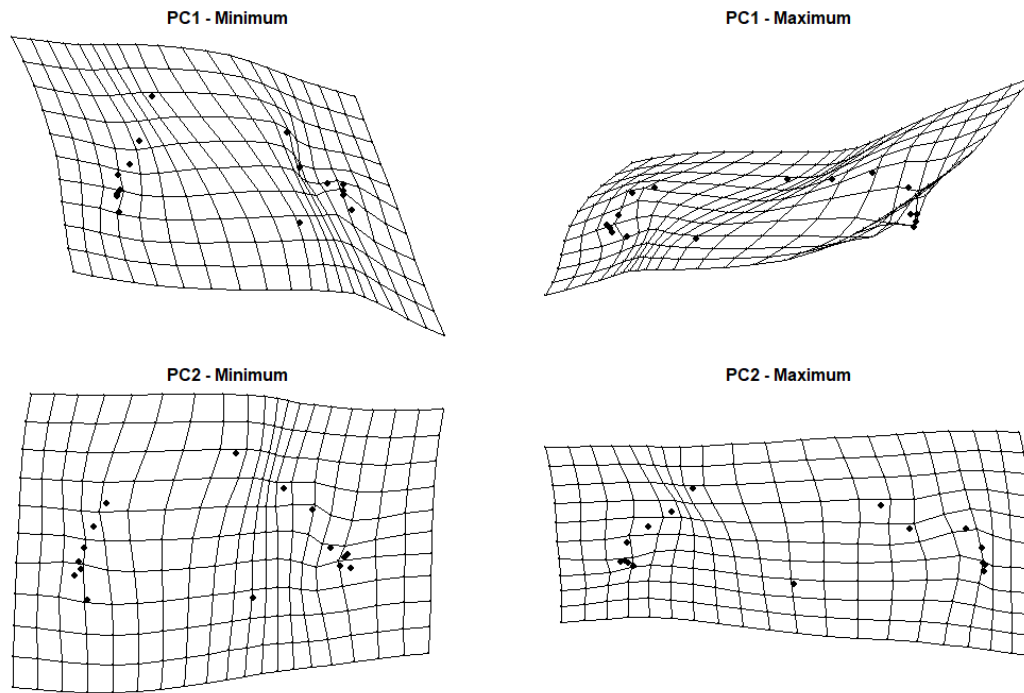


Figure 63: Semilandmarks (dots) and deformation grids in relation to consensus configuration (i.e. PC score = 0). Grids calculated using Procrustes coordinates acquired by SoundShape function. Data from the advertisement call of three populations of *Physalaemus deimaticus*. Upper panel: Minimum (left) and maximum (right) deformations from PC1 (39.7% variation). Lower panel: Minimum (left) and maximum (right) deformation from PC2 (24.6% variation).

The ordination of the first two Principal Components from SoundShape analysis (Figure 62) allied to the deformation grids for each PC (Figure 63) show clear structuring in the calls from different populations of *Physalaemus deimaticus*. Particularly for the population from Diamantina, the call is characterized by a round shape, scoring high values in PC1 and PC2 and resulting in the individuals from this locality structured at the top right hand corner of the ordination plot. On the other hand, the populations from Serra do Cipó and Pico do Itambé presented similar scores in both Components, resulting in these species structured together. In addition, the calls from Pico do Itambé encompassed great variation along both axis of the PCA, with scores that overlapped among all three localities.

Regarding the populations from Serra do Cipó and Diamantina, there is an apparent discrepancy in their sound shapes. Despite little overlap of scores at the center of ordination plot, these populations displayed clear structuring that matches the large geographical distance between them (*i.e.* approximate distance from Serra do Cipó to Diamantina: 110 km) in relation to the Pico do Itambé locality (*i.e.* approximate distance from Serra do Cipó to Pico do Itambé: 90 km), thus indicating that geographical separation is reflected as differences in sound shape between localities. In addition, the

similar scores between localities is expected and indicate that *Physalaemus deimaticus* has a similar sound shape along its distribution.

Also worth noting that the five notes analyzed for each individual often scored similar PC values. Although there is some variation among analyzed notes - especially in individuals from Diamantina - our results demonstrate that each individual of *Physalaemus deimaticus* has its own sound shape, which vary mainly around the PC1 axis (e.g. individuals “diamantina1”, “diamantina3”, “itambe1” and “itambe3”). The PC1 axis, in turn, is responsible for variation on the time X-axis (Figure 63), which imply that SoundShape likely encompassed variation in each notes duration, and it reflected on the PC scores and deformation grids.

```
# Plot ordination of Principal Components from eigensound
layout(matrix(c(1,1,2), 1, 3))
par(mgp=c(3,0.7,0))
pca.plot(PCA.out=eig.pca.deim, groups=deim.gr$Ind,
conv.hulls = deim.gr$Loc, leg=F, main=expression('eigensound PCA
- '*italic('P. deimaticus')* ' populations'))
plot(leg.interpop, axes=F) # Legend
```

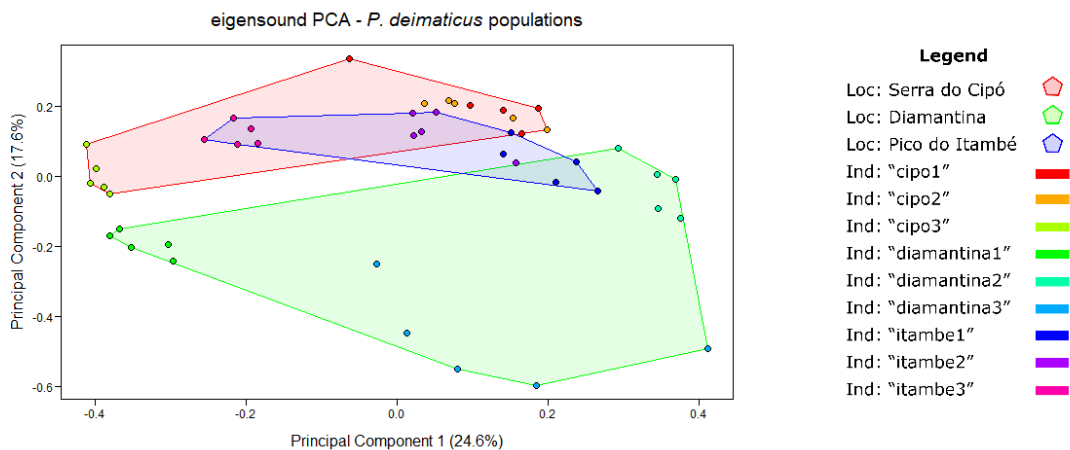


Figure 64: Individual projection of *Physalaemus deimaticus* individuals onto PC1 and PC2 axes using Procrustes coordinates acquired by `eigensound` function. Each point represent one sampled note and convex hulls comprise all notes analyzed for each population. Legend represent colors used for each individual and localities encompassed by convex hulls. Ind = individual. Loc = locality.

R code: Hypothetical semilandmark configurations for each Principal Component of PCA are calculated from Procrustes coordinates by the function `PCshapes` and plotted with `scatter3D` function.

```
# Acquire hypothetical semilandmark configurations from
Procrustes coordinates
```

```
eig.shapes.deim <- PCshapes(eig.gpa.deim)

# Hypothetical three dimensional semilandmark configurations
from PCs of eigensound analysis

par(mar=c(1, 1, 1, 1), mfrow=c(2,2))

scatter3D(x=eig.shapes.deim$PC1min[,1],
y=eig.shapes.deim$PC1min[,2],z=eig.shapes.deim$PC1min[,3],
theta=30, phi=35, resfac=1, r=3, expand=0.5, scale=T, axes=T,
cex.axis=0.7, ticktype="detailed", nticks=4, pch=21,
col=spectro.colors(n=100),cex=0.5, xlab="x", ylab="y", zlab="N",
clab=expression('Amplitude'), main=expression('PC1 - Minimum'))

scatter3D(x=eig.shapes.deim$PC1max[,1],
y=eig.shapes.deim$PC1max[,2],z=eig.shapes.deim$PC1max[,3],
theta=30, phi=35, resfac=1, r=3, expand=0.5, scale=T, axes=T,
cex.axis=0.7, ticktype="detailed", nticks=4, pch=21,
col=spectro.colors(n=100),cex=0.5, xlab="x", ylab="y", zlab="N",
clab=expression('Amplitude'), main=expression('PC1 - Maximum'))

scatter3D(x=eig.shapes.deim$PC2min[,1],
y=eig.shapes.deim$PC2min[,2],z=eig.shapes.deim$PC2min[,3],
theta=30, phi=35, resfac=1, r=3, expand=0.5, scale=T, axes=T,
cex.axis=0.7, ticktype="detailed", nticks=4, pch=21,
col=spectro.colors(n=100),cex=0.5, xlab="x", ylab="y", zlab="N",
clab=expression('Amplitude'), main=expression('PC2 - Minimum'))

scatter3D(x=eig.shapes.deim$PC2max[,1],
y=eig.shapes.deim$PC2max[,2],z=eig.shapes.deim$PC2max[,3],
theta=30, phi=35, resfac=1, r=3, expand=0.5, scale=T, axes=T,
cex.axis=0.7, ticktype="detailed", nticks=4, pch=21,
col=spectro.colors(n=100),cex=0.5, xlab="x", ylab="y", zlab="N",
clab=expression('Amplitude'), main=expression('PC2 - Maximum'))
```

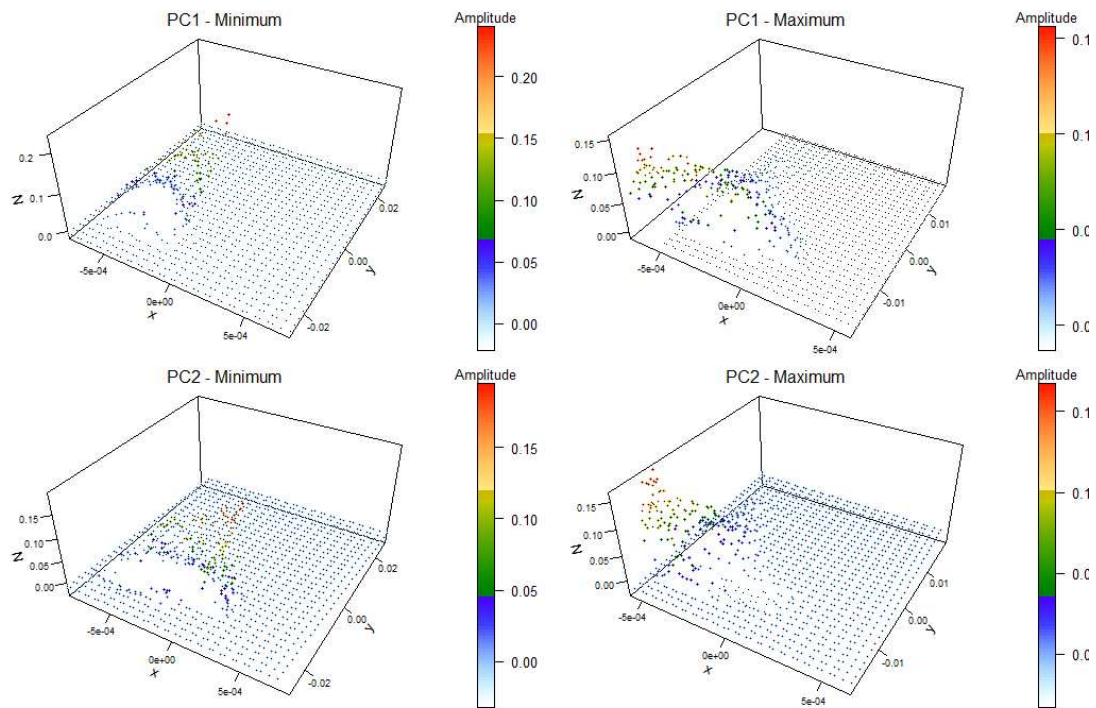


Figure 65: Hypothetical semilandmark configurations from minimum (left) and maximum (right) scores of PCA using Procrustes coordinates acquired by `eigensound` function. Data from the advertisement call of three populations of *Physalaemus deimaticus*. Upper panel: Minimum (left) and maximum (right) semilandmark configurations from PC1 (24.6% variation). Lower panel: Minimum (left) and maximum (right) semilandmark configurations from PC2 (17.6% variation).

Although the `eigensound` analysis resulted in less variation expressed by each Principal Component (Figure 64), the three dimensional approach yielded clearer structuring in the ordination of PCs from the call of *Physalaemus deimaticus* populations. Similarly as in the `SoundShape` analysis, the population from Diamantina displayed the greatest variation along both axis, which demonstrate great shape variation in the call from this locality. However, unlike the two dimensional approach, the populations from Diamantina and Serra do Cipó presented distinct PC scores that did not overlap, corroborating the accentuated differences in sound shape between allopatric populations. To that matter, Pico do Itambé population scored intermediate values in relation to the two distant populations, with sound shapes that vary along the PC1 axis.

As in the interspecific study, the hypothetical semilandmark configurations are not easily interpretable (Figure 65). The PC1 scores, for instance, appear to be related to the relative duration of each note along the sound window, whereas the PC2 scores reflect the bandwidth of each individuals call. The scores from Serra do Cipó, for instance, placed those individuals along the top of the ordination plot, from the left to the middle. That imply on a broad bandwidth (higher PC2 scores) with somewhat variable duration between individuals (low to medium PC1 scores).

Regarding an intrapopulation level, though, each individual scored similar values in both PCs, which led to structuring of individuals in the ordination plot (Figure 64). The exception is the individual “diamantina3”, which had variable notes placed along both PC axis. All other individuals had distinct scores for their calls, placing them along a very narrow space in the ordination plot. This indicates that the *eigensound* method managed to capture the each individual timbre in the calls of *Physalaemus deimaticus* populations.

Concluding remarks from empirical studies of SoundShape and eigensound methods

Both *SoundShape* and *eigensound* methods yielded clear structuring of the calls from *Physalaemus* species. *SoundShape* function simplifies spectrogram outputs into semilandmarks described by two dimensional point coordinates, which imply in a computationally less demanding method that results in higher statistical power in relation to the *eigensound* approach (MacLeod et al., 2013). However, the simplicity of *SoundShape* is also its main constraint, as it apparently misses some important features of each species calls, thus resulting in ordinations of Principal Components with groups of species that share similar calls rather than a group for each species. Still, *SoundShape* analysis enables the calculation of deformation grids in relation to the consensus shape (*i.e.* thin plates splines), which improve discussion on sound shape variation along the sample.

In general, *SoundShape* can be thought as a good method for grouping species based on sound shape similarity. We recommend its usage when one is looking for sonic similarities along a sample of organisms (*e.g.* shape comparison between monophyletic groups). In addition, *SoundShape* function served as a useful protocol for error verification in sound shape analysis, preventing errors from overlapping individuals or background noise in both *SoundShape* and *eigensound* approaches. The *eigensound* analysis, in turn, is very sensitive to the three dimensional call shapes. The method successfully encompassed the general structure of each *Physalaemus* call into a few Principal Components, which resulted in distinct scores for each species and clear groups in the ordination plot. Moreover, the function was accurate enough to result in three dimensional point coordinates that describe the timbre of each individual's call. Yet this high sensitivity is computationally very demanding, which prevents the creation of deformation grids for enhanced discussion of results from PCA. Finally, the hypothetical

semilandmark coordinates are often confuse and hard to interpret, not serving as well as the deformation grids from SoundShape.

Ultimately, the high sensitivity of eigensound analysis encompass a lot of variation from the calls, which remain as noise even after Generalized Procrustes Analysis. This variation spreads through several Principal Components in PCA and imply on a much lower statistical power in relation to SoundShape. Consequently, the first few PCs are only able to describe part of the variation from the dataset, meaning that several ordination plots would be necessary to comprehend how the three dimensional sound shape change along a sample. This is not ideal in multivariate analysis and might result in misinterpretation of results.

Despite these constraints, SoundShape and eigensound analysis stand as promising methods of bioacoustic analysis. SoundShape has the advantage of being faster, simpler, and very useful as a protocol for error verification to apply in the exploratory phases of both methods. eigensound, in turn, is a very accurate and sensitive method, being able to decompose the three dimensional spectrogram output into point coordinates that are unique to each species - and to each individual timbre. To that extent, our empirical study corroborate the results from MacLeod et al. (2013), in which the specificity of each species call is successfully described by the three dimensional approach. Although it is still early to draw conclusions on sound shape, it seems clear that there is a biological meaning on the shape of sound, which can be measured by either SoundShape or eigensound approaches.

References

- Adams, D. C., M. L. Collyer, A. Kaliontzopoulou, & E. Sherratt. (2017). *Geomorph: Software for geometric morphometric analyses*. R package version 3.0.5.
- Allaire, J. J., Xie, Y., McPherson, J., Luraschi, J., Ushey, K., Atkins, A., Wickham, H., Cheng, J. & Chang, W. (2017). *rmarkdown: Dynamic documents for R*. R package version 1.8. <https://CRAN.R-project.org/package=rmarkdown>
- Bookstein, F. L. (1991). *Morphometric tools for landmark data: Geometry and biology*. Cambridge University Press, New York.

- Bookstein, F. L. (1997). Landmark methods for forms without landmarks: morphometrics of group differences in outline shape. *Medical Image Analysis*, 1(3), 225-243.
- Cannatella, D. C., Hillis, D. M., Chippindale, P. T., Weigt, L., Rand, A. S., & Ryan, M. J. (1998). Phylogeny of frogs of the *Physalaemus pustulosus* species group, with an examination of data incongruence. *Systematic Biology*, 47(2), 311-35. [doi:10.1080/106351598260932](https://doi.org/10.1080/106351598260932)
- Cate, C. Ten. (2004). Birdsong and evolution. In *Nature's Music: The Science of Birdsong* (pp. 296-317). Elsevier Academic Press, London. [doi:10.1016/B978-012473070-0/50013-X](https://doi.org/10.1016/B978-012473070-0/50013-X)
- Charif, R. A., Waack, A. M., & Strickman, L. M. (2010). *Raven Pro 1.4 User's Manual*. Cornell Lab of Ornithology, Ithaca, NY.
- Crawley, M. J. (2012). *The R book (second edition)*. London: Wiley & Sons.
- Dryden, I. L., & Mardia, K. V. (1998). *Statistical shape analysis*. New York: John Wiley & Sons.
- Frontier, S. (1976). Étude de la décroissance des valeurs propres dans une analyse en composantes principales: Comparaison avec le modèle du bâton brisé. *Journal of Experimental Marine Biology and Ecology*, 25(1), 67-75. [doi:10.1016/0022-0981\(76\)90076-9](https://doi.org/10.1016/0022-0981(76)90076-9)
- Frost, D. R. (2017). *Amphibian Species of the World: an Online Reference*. Version 6.0 (DATE OF ACCESS). Retrieved from <http://research.amnh.org/herpetology/amphibia/index.html>
- Gerhardt, H. C. (1991). Female mate choice in treefrogs: Static and dynamic properties. *Animal Behaviour*, 42, 615-635.
- Gerhardt, H. C., & Huber, F. (2002). *Acoustic communication in insects and anurans: common problems and diverse solutions*. University of Chicago Press.
- Jackson, D. A. (1993). Stopping rules in Principal Components Analysis: A comparison of heuristical and statistical approaches. *Ecology*, 74(8), 2204-2214.
- Kendall, D. G. (1977). The diffusion of shape. *Advances in applied probability*, 9(3), 428-430.

- Köhler, J., Jansen, M., Rodriguez, A., Kok, P. J. R., Toledo, L. F., Emmrich, M., Glaw, F., Haddad, C. F. B., Rödel, M. O., & Vences, M. V. (2017). The use of bioacoustics in anuran taxonomy: theory, terminology, methods and recommendations for best practice. *Zootaxa*, 4251(1), 1-124. [doi:10.11646/zootaxa.4251.1.1](https://doi.org/10.11646/zootaxa.4251.1.1)
- Lourenço, L. B., Targueta, C. P., Baldo, D., Nascimento, J., Garcia, P. C. A., Andrade, G. V., Recco-Pimentel, S. M. (2015). Phylogeny of frogs from the genus *Physalaemus* (Anura, Leptodactylidae) inferred from mitochondrial and nuclear gene sequences. *Molecular Phylogenetics and Evolution*, 92, 204-16. [doi:10.1016/j.ympev.2015.06.011](https://doi.org/10.1016/j.ympev.2015.06.011).
- MacLeod, N., Krieger, J., & Jones, K. E. (2013). Geometric morphometric approaches to acoustic signal analysis in mammalian biology. *Hystrix, the Italian Journal of Mammalogy*, 24(1), 110-125. [doi:10.4404/hystrix-24.1-6299](https://doi.org/10.4404/hystrix-24.1-6299).
- Marler, P. (2004). Bird calls: A cornucopia for communication. In *Nature's Music: The Science of Birdsong* (pp. 132-177). Elsevier Academic Press, London. [doi:10.1016/B978-012473070-0/50008-6](https://doi.org/10.1016/B978-012473070-0/50008-6)
- McLister, J. D., Stevens, E. D., & Bogart, J. P. (1995). Comparative contractile dynamics of calling and locomotor muscles in three hylid frogs. *The Journal of Experimental Biology*, 198, 1527-1538.
- Päckert, M., Martens, J., Kosuch, J., Nazarenko, A. A., & Veith, M. (2003). Phylogenetic signal in the song of crests and kinglets (Aves: Regulidae). *Evolution*, 57(3), 616-629. [doi:10.1111/j.0014-3820.2003.tb01553.x](https://doi.org/10.1111/j.0014-3820.2003.tb01553.x)
- Podani, J., & Miklós, I. (2002). Resemblance coefficients and the Horseshoe Effect in Principal Coordinates Analysis. *Ecology*, 83(12), 3331-3343. [doi:10.1890/0012-9658\(2002\)083\[3331:RCATHE\]2.0.CO;2](https://doi.org/10.1890/0012-9658(2002)083[3331:RCATHE]2.0.CO;2)
- Prum, R. O. (1998). Sexual selection and the evolution of mechanical sound production in manakins (Aves: Pipridae). *Animal Behaviour*, 55, 977-994.
- R Core Team (2017). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, Austria. URL <https://www.R-project.org/>.
- Ragge, D. R., & Reynolds, W. J. (1998). *The songs of the Grasshoppers and Crickets of Western Europe*. Harley Books, Colchester.

- Robillard, T., & Desutter-Grandcolas, L. (2011). Evolution of calling songs as multicomponent signals in crickets (Orthoptera: Grylloidea: Eneopterinae). *Behaviour*, 148(5), 627-672. [doi:10.1163/000579511X572044](https://doi.org/10.1163/000579511X572044)
- Robillard, T., Höbel, G., & Gerhardt, H. C. (2006). Evolution of advertisement signals in North American hylid frogs: vocalizations as end-products of calling behavior. *Cladistics*, 22(6), 533-545. [doi:10.1111/j.1096-0031.2006.00118.x](https://doi.org/10.1111/j.1096-0031.2006.00118.x)
- Rohlf, F. J. (2015). The tps series of software. *Hystrix*, 26(1).
- Rohlf, F. J., & Slice, D. (1990). Extensions of the Procrustes method for the optimal superimposition of landmarks. *Systematic Biology*, 39(1), 40-59.
- Schnitzler, H.-U., & Kalko, E. K. V. (2001). Echolocation by insect-eating bats. *BioScience*, 51(7), 557-569. [doi:10.1641/0006-3568\(2001\)051\[0557:EBIEB\]2.0.CO;2](https://doi.org/10.1641/0006-3568(2001)051[0557:EBIEB]2.0.CO;2)
- Sheets, H. D., Covino, K. M., Panasiewicz, J. M., & Morris, S. R. (2006). Comparison of geometric morphometric outline methods in the discrimination of age-related differences in feather shape. *Frontiers in Zoology*, 3(15), 1-12. [doi:10.1186/1742-9994-3-15](https://doi.org/10.1186/1742-9994-3-15)
- Sherratt, E. (2017). *Quick Guide to geomorph 3.0.5*. Available from: http://www.public.iastate.edu/~dcadams/PDFPubs/Quick_Guide_to_Geomorph-3.0.pdf
- Zelditch, M. L., Swiderski, D. L., Sheets, H. D., & Fink, W. L. (2012). *Geometric morphometrics for biologists: A primer. Elsevier (Second Edition)*. Elsevier, San Diego. [doi:10.1016/B978-0-12-386903-6.00001-0](https://doi.org/10.1016/B978-0-12-386903-6.00001-0)

14 – Appendix 3: SoundShape function: Computing semilandmarks from a curve of relative amplitude. For details on arguments of the function, please see Appendix 2.

```
SoundShape <- function(f=44100, wl=512, ovlp=50, dBlevel=25,
  flim=c(0, 6), tlim=c(0, 4), trel=c(0, 4), mag.time=1,
  wav.at = getwd(), store.at = getwd(),EQ= c(0.05, 0.15, 0.3,
  0.5, 0.7, 0.85, 0.95), plot.exp=TRUE, plot.as="jpeg",
  TPS.file=NULL){

  if(!library("seewave",
logical.return=T)){install.packages("seewave")}

  if(!library("tuneR",
logical.return=T)){install.packages("tuneR")}

  if(!library("abind",
logical.return=T)){install.packages("abind")}

  require("seewave")

  require("tuneR")

  require("abind")

  ZC <- function(x) {

    sign <- ifelse(x >= 0, 1, -1)

    res <- abs(diff(sign))

    return(res)  }

  frequencies <- function (spec, f = NULL, mel = FALSE, flim
= NULL) {

    fhz <- f

    if (!is.null(f) & mel) {f <- 2 * mel(f/2)}

    if (is.null(f)) {

      if (is.vector(spec))

        stop("'f' is missing")

      else if (is.matrix(spec))
```

```

        f      <-      spec[nrow(spec),      1]      *      2000      *
nrow(spec)/(nrow(spec) - 1)      }

    if (is.matrix(spec)) {

        freq <- spec[, 1]

        freq = freq * 1000

        spec <- spec[, 2] }

L <- length(spec)

wl <- L * 2

if (any(spec < 0))

    stop("The frequency spectrum to be analysed should not
be in dB")

if (!is.null(flim)) {

    if (flim[1] < 0 || flim[2] > fhz/2)

        stop("'flim' should range between 0 and f/2")

    if (mel)

        flim <- mel(flim * 1000)/1000 }

else {flim <- c(0, f/2000)}

spec <- spec[(flim[1] * 1000 * wl/f):(flim[2] * 1000 *
wl/f)]

amp <- spec/sum(spec)

cumamp <- cumsum(amp)

freq <- seq(from = flim[1] * 1000, to = flim[2] * 1000,
length.out = length(spec))

for(i in EQ){if(!exists("Qlist")){Qlist <- data.frame()}

    if(exists("Qlist")) {

```

```

      Qtemp <- data.frame("Quantiles" = paste("EQ-", i,
sep=""), "Frequency values" = freq[length(cumamp[cumamp <=
i]) + 1])

      Qlist <- rbind(Qlist, Qtemp)

      rm(Qtemp) } }

results <- Qlist }

if(!is.null(TPS.file)){

  TPS.path <- paste(store.at, "/", TPS.file, ".tps",
sep="")

  file.create(TPS.path, showWarnings = TRUE) }

for(file in list.files(wav.at, pattern = ".wav")) {

  Wav <- readWave(paste(wav.at, "/", file, sep=""))

  s <- spectro(Wav, f=f, wl=wl, ovlp=ovlp, flim=flim,
tlim=tlim, plot=FALSE)

  con <- contourLines(x=s$time, y=s$freq, z=t(s$amp),
levels=seq(-dBlevel, -dBlevel, 1))

  n <- numeric(length(con))

  for(i in 1:length(con)) n[i] <- length(con[[i]]$x)

  n.max <- which.max(n)

  con <- con[[n.max]]

  t.beg <- min(con$x)

  t.end <- max(con$x)

  f.min <- min(con$y)

  t.min <- con$x[con$y==f.min]

  f.max <- max(con$y)

  t.max <- con$x[con$y==f.max]

```

```

pwr.spec <- spec(Wav, f=f, wl=wl, from=t.beg, to=t.end,
plot=F)

fvalues <- frequencies(pwr.spec, f=f, flim=c(f.min,
f.max))

fvalues$Frequency.values <-
(fvalues$Frequency.values)/1000

for(i in seq_along(fvalues$Frequency.values)) {

  zc.res <- ZC(con$y - fvalues$Frequency.values[i])

  t.pos <- which(zc.res==2)

  if(!exists("t1")||!exists("t2")){

    t1<-vector(mode="integer",
length=length(fvalues$Frequency.values))

    t2<-vector(mode="integer",
length=length(fvalues$Frequency.values)) }

  if(exists("t1")|exists("t2")){

    t1[i] <- min(con$x[t.pos])

    t2[i] <- max(con$x[t.pos]) } }

SM1 <- data.frame(cbind(fvalues, "Time"= t1))

SM2 <- data.frame(cbind(fvalues, "Time"= t2))

SM1 <- SM1[order(SM1[,2], decreasing=T), ]

SM.min <- data.frame("Quantiles" = "EQ-min",
"Frequency.values"= f.min, "Time"= t.min)

SM.max <- data.frame("Quantiles" = "EQ-max",
"Frequency.values"= f.max, "Time"= t.max)

SM.numbers <- as.factor(1:((length(EQ)*2) + 2)) # Twice
the length of EQ plus the SM-min and SM-max

SM <- cbind(SM.numbers, data.frame(rbind(SM.max, SM1,
SM.min, SM2)))

```

```

SM[,4] <- SM[,4]*mag.time

SM.mtx <- as.matrix(SM[,4:3])

ifelse(!exists('coordSM'),

      coordSM <- array(data=SM.mtx, dim=c(dim(SM.mtx), 1)),

      coordSM <- abind::abind(coordSM, SM.mtx, along=3))

if(plot.exp == TRUE){

  if(plot.as == "jpeg"){jpeg(width =5000,height = 3500,
units = "px", res = 500,filename=paste(store.at,"/",
sub(".wav","",file), ".jpg", sep=""))}

  if(plot.as=="tiff"|plot.as=="tif"){tiff(width=5000,
height=3500, units="px", res=500, filename=paste(store.at,
"/", sub(".wav", "", file), ".tif", sep=""))}
spectro(Wav, f=f, wl=wl, ovlp=ovlp, osc=F, scale=F, grid=F,
main= sub(".wav", "", file), trel=trel, flim=flim,
tlim=tlim)

  polygon(x=con$x, y=con$y, lwd=1.5)

  points(x=(SM$Time)/mag.time, y=SM$Frequency.values,
pch=21, col="black", bg="red", cex=1.1)

  text(x=(SM$Time)/mag.time, y=SM$Frequency.values,
cex=0.7, labels=SM$SM.numbers, pos=c(3, rep(2, dim(SM1)[1]),
1, rep(4, dim(SM2)[1])))

  dev.off() }

if(!is.null(TPS.file)){

  lmline <- paste("LM=", dim(SM)[1], sep="")

  write(lmline, TPS.path, append = TRUE)

  write.table(SM[,4:3], TPS.path, col.names = FALSE,
row.names = FALSE, append = TRUE)

  idline <- paste("ID=", sub(".wav", "", file), sep="")

  write(idline, TPS.path, append = TRUE)

```

```
write("", TPS.path, append = TRUE)

rm(lmline, idline, SM) } }

dimnames(coordSM)[[3]] <- sub(".wav","", list.files(wav.at,
pattern = ".wav"))

results <- coordSM }
```

15 – Appendix 4: eigensound function: Computing semilandmarks from the three dimensional surface of sound. For details on arguments of the function, please see Appendix 2.

```
eigensound <- function(f=44100, wl=512, ovlp=50, dBlevel=25,
  flim=c(0, 6), tlim = c(0, 4), x.length=100, y.length=80,
  wav.at = getwd(), store.at = getwd(), plot.exp = TRUE,
  plot.as = "jpeg", plot.type = "surface", TPS.file = NULL,
  log.scale=TRUE){

  if(!library("reshape2",
logical.return=T)){install.packages("reshape2")}

  if(!library("seewave",
logical.return=T)){install.packages("seewave")}

  if(!library("tuneR",
logical.return=T)){install.packages("tuneR")}

  if(!library("abind",
logical.return=T)){install.packages("abind")}

  if(!library("plot3D",
logical.return=T)){install.packages("plot3D")}

  require("reshape2")

  require("seewave")

  require("tuneR")

  require("abind")

  require("plot3D")

  if(!is.null(TPS.file)){

    TPS.path <- paste(store.at, "/", TPS.file, ".tps",
sep="")

    file.create(TPS.path, showWarnings = TRUE)  }

  for(file in list.files(wav.at, pattern = ".wav")) {

    eig <- readWave(paste(wav.at, "/", file, sep=""))
```

```

e <- spectro(eig, f=f, wl=wl, ovlp=ovlp, flim=flim,
tlim=tlim, plot=F)

freq.seq <- seq(1, length(e$freq), length = y.length)

ifelse(isTRUE(log.scale), time.seq <- 10^(seq(log10(1),
log10(length(e$time)), length.out = x.length)), time.seq <-
seq(1, length(e$time), length.out = x.length))

time.sub <- e$time[time.seq]

freq.sub <- e$freq[freq.seq]

amp.sub <- e$amp[freq.seq, time.seq]

for(i in 1:length(amp.sub)){if(amp.sub[i] == -Inf
|amp.sub[i] <= -dBlevel) {amp.sub[i] <- -dBlevel}}

colnames(amp.sub) <- time.sub

rownames(amp.sub) <- freq.sub

ind.3D <- as.matrix(setNames(reshape2::melt(t(amp.sub)),
c('time', 'freq', 'amp'))))

ifelse(!exists('coord3D'), coord3D <- array(data=ind.3D,
dim=c(dim(ind.3D), 1)), coord3D <- abind::abind(coord3D,
ind.3D, along=3))

if(plot.exp==TRUE){

  if(plot.as == "jpeg"){jpeg(width =5000,height = 3500,
units = "px", res = 500, filename=paste(store.at,"/",
sub(".wav","",file), ".jpg", sep=""))}

  if(plot.as=="tiff"|plot.as=="tif"){tiff(width=5000,
height=3500, units="px", res=500, filename=paste(store.at,
"/", sub(".wav", "", file), ".tif", sep=""))}

  if(plot.type=="surface"){persp3D(x=time.sub,
y=freq.sub, z=t(amp.sub), border="black", lwd=0.1, theta=30,
phi=35, resfac=1, r=3,expand=0.5, cex.axis=0.7, scale=T,

```

```

axes=T,    col=spectro.colors(n=100),    ticktype="detailed",
nticks=4,    xlab="Time    (s)",    ylab="Frequency    (kHz)",
zlab="Amplitude    (dB)",    main=    sub(".wav",    "",    file),
clab=expression('Amplitude dB'))}

    if(plot.type=="points"){scatter3D(x=ind.3D[,1],
y=ind.3D[,2],    z=ind.3D[,3],    pch=21,    cex=0.5,    theta=30,
phi=35,    resfac=1,    r=3,    expand=0.5,    cex.axis=0.7,    scale=T,
axes=T,    col=spectro.colors(n=100),    ticktype="detailed",
nticks=4,    xlab="Time    (s)",    ylab="Frequency    (kHz)",
zlab="Amplitude    (dB)",    main=    sub(".wav",    "",    file),
clab=expression('Amplitude dB'))}

    dev.off()    }

if(!is.null(TPS.file)){

    lmline <- paste("LM=", dim(ind.3D)[1], sep="")

    write(lmline, TPS.path, append = TRUE)

    write.table(ind.3D,    TPS.path,    col.names    =    FALSE,
row.names = FALSE, append =TRUE)

    idline <- paste("ID=", sub(".wav", "", file), sep="")

    write(idline, TPS.path, append = TRUE)

    write("", TPS.path, append = TRUE)

    rm(lmline, idline)    }    }

    dimnames(coord3D)[[3]]<-sub(".wav","", list.files(wav.at,
pattern = ".wav"))

    results <- coord3D    }

```