

# Package ‘zoolog’

September 2, 2021

**Title** Zooarchaeological Analysis with Log-Ratios

**Version** 0.4.1

**Date** 2021-8-31

**Description** Includes functions and reference data to generate and manipulate log-ratios (also known as log size index (LSI) values) from measurements obtained on zooarchaeological material. Log ratios are used to compare the relative (rather than the absolute) dimensions of animals from archaeological contexts (Meadow 1999, ISBN: 9783896463883). zoolog is also able to seamlessly integrate data and references with heterogeneous nomenclature, which is internally managed by a zoolog thesaurus. A preliminary version of the zoolog methods was first used by Trentacoste, Nieto-Espinet, and Valenzuela-Lamas (2018) <[doi:10.1371/journal.pone.0208109](https://doi.org/10.1371/journal.pone.0208109)>.

**License** GPL-3

**Encoding** UTF-8

**Language** en-GB

**LazyData** true

**RoxygenNote** 7.1.1

**Suggests** testthat, knitr, rmarkdown, ggplot2, stats

**Depends** R (>= 3.0)

**Imports** utils, stringi, Rdpack (>= 0.7)

**RdMacros** Rdpack

**VignetteBuilder** knitr

**URL** <https://josempozo.github.io/zoolog/>

**NeedsCompilation** no

**Author** Jose M Pozo [aut, cre] (<<https://orcid.org/0000-0002-0759-3510>>),  
Silvia Valenzuela-Lamas [aut] (<<https://orcid.org/0000-0001-9886-0372>>),  
Angela Trentacoste [aut] (<<https://orcid.org/0000-0001-7096-5252>>),  
Ariadna Nieto-Espinet [aut] (<<https://orcid.org/0000-0003-2567-1735>>),

Silvia Guimarães Chiarelli [aut]  
 (<<https://orcid.org/0000-0002-3778-3315>>)

**Maintainer** Jose M Pozo <josmpozo@gmail.com>

**Repository** CRAN

**Date/Publication** 2021-09-02 06:40:02 UTC

## R topics documented:

AssembleReference . . . . .	2
CondenseLogs . . . . .	3
dataValenzuelaLamas2008 . . . . .	5
InCategory . . . . .	6
LogRatios . . . . .	7
referencesDatabase . . . . .	10
RemoveNACases . . . . .	16
StandardizeNomenclature . . . . .	17
ThesaurusManagement . . . . .	19
ThesaurusReaderWriter . . . . .	21
zoologThesaurus . . . . .	22
<b>Index</b>	<b>25</b>

---

AssembleReference	<i>Assemble Reference</i>
-------------------	---------------------------

---

## Description

Function to build a reference dataframe selecting a case for each taxon from the available specimens in the references' database.

## Usage

```
AssembleReference(
  combination,
  ref.db = referencesDatabase,
  thesaurus = zoologThesaurus$taxon
)
```

## Arguments

combination	A dataframe or named list. Each (column) name identifies a taxon. Each column or list element must have a single element of type character, identifying one of the sources included in the references' database.
ref.db	A reference database. This is a named list of named lists of dataframes. The first level is named by taxon and the second level is named by reference source. Each dataframe includes the reference for the corresponding taxon and source. The default ref.db = <a href="#">referencesDatabase</a> is provided as package <b>zoolog</b> data.

thesaurus      A thesaurus for taxa.

### Value

A reference dataframe.

### Examples

```
## `referenceSets` includes a series of predefined reference compositions.
referenceSets
## Actually the package `references` is build from them.
## We can rebuild any of them:
referenceCombi <- AssembleReference(referenceSets["Combi", ])

## Define an alternative reference combining differently the references'
## database:
refComb <- list(cattle = "Nieto", sheep = "Davis", Goat = "Clutton",
               pig = "Albarella", redDeer = "Basel")
userReference <- AssembleReference(refComb)
```

---

CondenseLogs

*Condense Measure Log-Ratios*

---

### Description

This function condenses the calculated log ratio values into a reduced number of features by grouping log ratio values and selecting or calculating a feature value. By default the selected groups each represents a single dimension, i.e. Length and Width. Only one feature is extracted per group. Currently, two methods are possible: priority (default) or average.

### Usage

```
CondenseLogs(
  data,
  grouping = list(Length = c("GL", "GL1", "GLm", "HTC"), Width = c("BT", "Bd", "Bp",
    "SD", "Bfd", "Bfp")),
  method = "priority"
)
```

### Arguments

data	A dataframe with the input measurements.
grouping	A list of named character vectors. The list includes a vector per selected group. Each vector gives the group of measurements in order of priority. By default the groups are Length = c("GL", "GL1", "GLm", "HTC") and Width = c("BT", "Bd", "Bp", "SD", "Bfd", "Bfp"). The order is irrelevant for method = "average".
method	Character string indicating which method to use for extracting the condensed features. Currently accepted methods: "priority" (default) and "average".

## Details

This operation is motivated by two circumstances. First, not all measurements are available for every bone specimen, which obstructs their direct comparison and statistical analysis. Second, several measurements can be strongly correlated (e.g. *SD* and *Bd* both represent bone width). Thus, considering them as independent would produce an over-representation of bone remains with more measurements per axis. Condensing each group of measurements into a single feature (e.g. one measure per axis) palliates both problems.

Observe that an important property of the log-ratios from a reference is that it makes the different measures comparable. For instance, if a bone is scaled with respect to the reference, so that it homogeneously doubles its width, then all width related measures (*BT*, *Bd*, *Bp*, *SD*, ...) will give the same log-ratio ( $\log(2)$ ). In contrast, the absolute measures are not directly comparable.

The measurement names in the grouping list are given without the `logPrefix`. But the selection is made from the log-ratios.

The default method is "priority", which selects the first available measure log-ratio in each group. The method "average" extracts the mean per group, ignoring the non-available measures. We provide the following by-default group and prioritization: For lengths, the order of priority is: *GL*, *GLl*, *GLm*, *HTC*. For widths, the order of priority is: *BT*, *Bd*, *Bp*, *SD*, *Bfd*, *Bfp*. This order maximises the robustness and reliability of the measurements, as priority is given to the most abundant, more replicable, and less age dependent measurements.

This method was first used in: Trentacoste, A., Nieto-Espinet, A., & Valenzuela-Lamas, S. (2018). Pre-Roman improvements to agricultural production: Evidence from livestock husbandry in late prehistoric Italy. *PloS one*, 13(12), e0208109.

Alternatively, a user-defined method can be provided as a function with a single argument (`data.frame`) assumed to have as columns the measure log-ratios determined by the grouping.

## Value

A dataframe including the input dataframe and additional columns, one for each extracted condensed feature, with the corresponding name given in grouping.

## Examples

```
## Read an example dataset:
dataFile <- system.file("extdata", "dataValenzuelaLamas2008.csv.gz",
                        package="zoolog")
dataExample <- utils::read.csv2(dataFile,
                                na.strings = "",
                                encoding = "UTF-8",
                                stringsAsFactors = TRUE)

## Compute the log-ratios and select the cases with available log ratios:
dataExampleWithLogs <- RemoveNACases(LogRatios(dataExample))
## We can observe the first lines (excluding some columns for visibility):
head(dataExampleWithLogs)[, -c(6:20,32:63)]

## Extract the default condensed features with the default "priority" method:
dataExampleWithSummary <- CondenseLogs(dataExampleWithLogs)
head(dataExampleWithSummary)[, -c(6:20,32:63)]
```

```
## Extract only width with "average" method:
dataExampleWithSummary2 <- CondenseLogs(dataExampleWithLogs,
                                         grouping = list(Width = c("BT", "Bd", "Bp", "SD")),
                                         method = "average")
head(dataExampleWithSummary2)[, -c(6:20, 32:63)]
```

---

dataValenzuelaLamas2008

*Example dataset*

---

## Description

The dataset provided as an example originates from (Valenzuela-Lamas 2008). The dataset is written in Catalan, with the exception of some headings to facilitate understanding of its contents.

## Format

The dataset is provided in the **zoolog** extdata folder as a file in semicolon-separated values format but compressed with *gzip* to reduce its size:

dataValenzuelaLamas2008.csv.gz

The file is provided in UTF-8 encoding. The file encoding is relevant because the dataset contains accents and special characters that needs to be correctly displayed. It can be directly open by `utils::read.csv2`, provided that the correct encoding is set (see examples below).

Every row of the data.frame refers to one individual bone fragment unless otherwise stated in the *Observations* field ("Observacions").

All the measurements are expressed in millimetres and were obtained with a manual calliper.

The main headings in the database are:

**Site** The faunal remains from three Iron Age archaeological sites were recorded (ALP = Alorda Park, TFC = Turó de la Font de la Canya, OLD = Olèrdola).

**N inv** A correlative number for each fragment.

**UE** Refers to the Stratigraphic Unit (SU in English).

**Especie** Refers to the species.

**Os** Refers to the skeletal element.

**Fragment** Refers to the preserved part in the vertical axis (distal, proximal, diaphysis, etc.).

**Lat** Bone laterality: right (d) or left (e).

**Vora** Refers to the preserved part in relation to the circumference (c), or a vertically, transversally and obliquely fragmented (sto).

**Fract** Refers to fracture during field excavation or lab work.

**Tafo** Refers to anthropic and post-depositional alterations.

**Grau** Refers to degree of bone alteration in a scale from 0 (no alteration) to 4 (diaphysis completely altered).

**Epif** Degree of fusion: s= fused, ns= unfused, ec = fusion visible. Also tooth wear is recorded here following (Gardeisen 1997).

**Sexe** Sex: male (masc) / female (fem).

**Traces** Refers to butchery marks. It may also include other observations.

**Observacions** Observations.

**Recinte** Refers to the number of silo structure (e.g. SJ8) or the room (e.g. AB) from which the material originates.

**TPQ** Absolute chronology in Terminus Post Quem.

**TAQ** Absolute chronology in Terminus Ante Quem.

**Period** Chronological phasing.

**Capsa** Box number that contains the item.

**Measurement codes** The nomenclature follows (Von den Driesch 1976).

## References

Gardeisen A (1997). "Exploitation des prélèvements et fichiers de spécialité (PRL, FAUNE, OS)." *Lattara*, **10**, 251–278.

Valenzuela-Lamas S (2008). *Alimentació i ramaderia al Penedès durant la protohistòria (segles VII-III aC)*. Societat Catalana d'Arqueologia (Premi d'Arqueologia - Memorial Josep Barberà i Farràs, 5a edició). [http://www.scarqueologia.com/?page\\_id=10](http://www.scarqueologia.com/?page_id=10).

Von den Driesch A (1976). *A guide to the measurement of animal bones from archaeological sites: as developed by the Institut für Palaeoanatomie, Domestikationsforschung und Geschichte der Tiermedizin of the University of Munich*, volume 1. Peabody Museum Press.

## Examples

```
dataFile <- system.file("extdata", "dataValenzuelaLamas2008.csv.gz",
  package="zoolog")
dataExample <- utils::read.csv2(dataFile,
  na.strings = "",
  encoding = "UTF-8",
  stringsAsFactors = TRUE)
```

---

InCategory

*Value Matching by Thesaurus Category*

---

## Description

Function to check if an element belongs to a category according to a thesaurus. It is similar to `%in%` and `is.element`, returning a logical vector indicating if each element in a given vector is included in a given set. But `InCategory` checks for equality assuming the equivalencies defined in the given thesaurus.

**Usage**

```
InCategory(x, category, thesaurus)
```

**Arguments**

x	Character vector to be checked for its inclusion in the category.
category	Character vector identifying the categories in which the inclusion of x will be checked. Each category can be identified by any equivalent name in the thesaurus.
thesaurus	A thesaurus object.

**Value**

A logical vector of the same length as x. Each value answers the question: *Does the corresponding element in x belong to any of the thesaurus categories identified by category?*

**See Also**

[zoologThesaurus,%in%](#)

**Examples**

```
InCategory(c("sheep", "cattle", "goat", "red deer"),
           c("ovis", "capra"),
           zoologThesaurus$taxon)
```

---

LogRatios

*Log Ratios of Measurements*

---

**Description**

Function to compute the (base 10) log ratios of the measurements relative to standard reference values. By default a reference is provided with the package.

**Usage**

```
LogRatios(
  data,
  ref = reference$Combi,
  identifiers = c("TAX", "EL"),
  refMeasuresName = "Measure",
  refValuesName = "Standard",
  thesaurusSet = zoologThesaurus,
  joinCategories = NULL,
  mergedMeasures = NULL
)
```

### Arguments

<code>data</code>	A dataframe with the input measurements.
<code>ref</code>	A dataframe including the measurement values used as references. The default <code>ref = reference\$Combi</code> provided as package <b>zoolog</b> data.
<code>identifiers</code>	A vector of column names in <code>ref</code> identifying a type of bone. By default <code>identifiers = c("TAX", "EL")</code> .
<code>refMeasuresName</code>	The column name in <code>ref</code> identifying the type of bone measurement.
<code>refValuesName</code>	The column name in <code>ref</code> giving the measurement value.
<code>thesaurusSet</code>	A thesaurus allowing datasets with different nomenclatures to be merged. By default <code>thesaurusSet = zoologThesaurus</code> .
<code>joinCategories</code>	A list of named character vectors. Each vector is named by a category in the reference and includes a set of categories in the data for which to compute the log ratios with respect to that reference. When NULL (default) no grouping is considered.
<code>mergedMeasures</code>	A list of character vectors or a single character vector. Each vector identifies a set of measures that the data presents merged in the same column, named as any of them. This practice only makes sense if only one of the measures can appear in each bone element.

### Details

Each log ratio is defined as the decimal logarithm of the ratio of the variable of interest to a corresponding reference value.

The `identifiers` are expected to determine corresponding columns in both data and reference. Each value in these columns identifies the type of bone. By default this is determined by a taxon and a bone element. For any case in the data, the log ratios are computed with respect to the reference values in the same bone type. If the reference does not include that bone type, the corresponding log ratios are set to NA.

For some applications it can be interesting to group some set of bone types into the same reference category to compute the log ratios. The parameter `joinCategories` allows this grouping. `joinCategories` must be a list of named vectors, each including the set of categories in the data which should be mapped to the reference category given by its name.

This can be applied to group different species into a single reference species. For instance *sheep*, *capra*, and doubtful cases between both (*sheep/capra*), can be grouped and matched to the same reference for *sheep*, by setting `joinCategories = list(sheep = c("sheep", "capra", "oc"))`. Similarly, it can be applied to group different bone elements into a single reference (see the example below for undetermined phalanges).

Note that the `joinCategories` option does not remove the distinction between the different bone types in the data, just indicates that for any of them the log ratios must be computed from the same reference.

There are some measures that are restricted to a subset of bones. For instance, *GII* is only relevant for the *astragalus*, while *GL* is not applicable to it. Thus, there cannot be any ambiguity between both measures since they can be identified by the bone element. This justifies that some users have simplified datasets where a single column records indistinctly *GL* or *GII*. The optional parameter



mergedMeasures facilitates the processing of this type of simplified dataset. For the alluded example, mergedMeasures = list(c("GL", "GL1")) automatically selects, for each bone element, the corresponding measure present in the reference.

Observe that if mergedMeasures is set to non mutually exclusive measures, the behaviour is unpredictable.

## Value

A dataframe including the input dataframe and additional columns, one for each extracted log ratio for each relevant measurement in the reference. The name of the added columns are constructed by prefixing each measurement by the internal variable logPrefix.

If the input dataframe includes additional S3 classes (such as "tbl\_df"), they are also passed to the output.

## Examples

```
## Read an example dataset:
dataFile <- system.file("extdata", "dataValenzuelaLamas2008.csv.gz",
                        package="zoolog")
dataExample <- utils::read.csv2(dataFile,
                               na.strings = "",
                               encoding = "UTF-8",
                               stringsAsFactors = TRUE)

## For illustration purposes we keep now only a subset of cases to make
## the example run sufficiently fast.
## Avoid this step if you want to process the full example dataset.
dataExample <- dataExample[145:1000, ]
## We can observe the first lines (excluding some columns for visibility):
head(dataExample)[, -c(6:20,32:64)]

## Compute the log-ratios with respect to the default reference in the
## package zoolog:
dataExampleWithLogs <- LogRatios(dataExample)
## The output data frame include new columns with the log-ratios of the
## present measurements, in both data and reference, with a "log" prefix:
head(dataExampleWithLogs)[, -c(6:20,32:64)]

## Compute the log-ratios with respect to a different reference:
dataExampleWithLogs2 <- LogRatios(dataExample, ref = reference$Basel)
head(dataExampleWithLogs2)[, -c(6:20,32:64)]

## Define an alternative reference combining differently the references'
## database:
refComb <- list(cattle = "Nieto", sheep = "Davis", Goat = "Clutton",
              pig = "Albarella", redDeer = "Basel")
userReference <- AssembleReference(refComb)
## Compute the log-ratios with respect to this alternative reference:
dataExampleWithLogs3 <- LogRatios(dataExample, ref = userReference)

## We can be interested in including the first and second phalanges without
## anterior-posterior identification ("phal 1" and "phal 2"), by computing
```

```
## their log ratios with respect to the reference of the corresponding
## anterior first phalanges ("phal 1 ant" and "phal 2 ant", respectively).
## For this we use the optional argument joinCategories:
categoriesPhalAnt <- list('phal 1 ant' = c("phal 1 ant", "phal 1"),
                        'phal 2 ant' = c("phal 2 ant", "phal 2"))
dataExampleWithLogs4 <- LogRatios(dataExample,
                                joinCategories = categoriesPhalAnt)
head(dataExampleWithLogs4)[, -c(6:20,32:64)]
```

---

referencesDatabase      *References*

---

### Description

Several osteometrical references are provided in **zoolog** to enable researchers to use the one of their choice. The user can also use their own osteometrical reference if preferred.

### Usage

reference

referenceSets

referencesDatabase

### Format

Each reference is a data.frame including 4 columns:

**TAX** The taxon to which each reference bone belongs.

**EL** The skeletal element.

**Measure** The type of measurement taken on the bone.

**Standard** The value of the measurement taken on the bone. All the measurements are expressed in millimetres.

### Data Source

Currently, the references include reference values for the main domesticates and their agriotypes (*Bos*, *Ovis*, *Capra*, *Sus*), and other less frequent species, such as red deer and donkey, drawn from the following publications and resources:

**Cattle - Bos Nieto** *Bos taurus*. Female cow dated to the Early Bronze Age (Minferri, Catalonia), in Nieto-Espinet (2018).

**Basel** *Bos taurus*. Inv.nr. 2426 (Hinterwälder; female; 17 years old; live weight: 340 kg; withers height: 113 cm), from Stopp and Deschler-Erb (2018).

**Johnstone** *Bos taurus*. Standard values from means of cattle measures from Period II (Late Iron Age to Romano-British transition) of Elms Farm, Heybridge (Johnstone and Albarella 2002).

- Degerbøl** *Bos primigenius*. Female aurochs from Degerbøl and Fredskild (1970).
- Sheep - Ovis Davis** *Ovis aries*. Mean values of measurements from a group of adult female Shetland sheep skeletons from a single flock (Davis 1996).
- Clutton** *Ovis aries*. Mean measurements from a group of male Soay sheep of known age (Clutton-Brock et al. 1990).
- Basel** *Ovis musimon*. Inv.nr. 2266 (male; adult), from Stopp and Deschler-Erb (2018).
- Uerpmann** *Ovis orientalis*. Field Museum of Chicago catalogue number: FMC 57951 (female; western Iran) from Uerpmann and Uerpmann (1994).
- Goat - Capra Basel** *Capra hircus*. Inv.nr. 1597 (male; adult), from Stopp and Deschler-Erb (2018).
- Clutton** *Capra hircus*. Mean measurements from a group of goats of unknown age and sex (Clutton-Brock et al. 1990).
- Uerpmann** *Capra aegagrus*. Measurements based on female and male *Capra aegagrus*, Natural History Museum in London number: BMNH 651 M and L2 (Taurus Mountains in southern Turkey) from Uerpmann and Uerpmann (1994).
- Pig - Sus Albarella** *Sus domesticus*. Mean measurements from a group of Late Neolithic pigs from Durrington Walls, England (Albarella and Payne 2005).
- Basel** *Sus scrofa*. Inv.nr. 1446 (male; 2-3 years old; life weight: 120 kg) from Stopp and Deschler-Erb (2018).
- Hongo** *Sus scrofa*. Averaged left and right measurements of a female wild boar from near Elaziğ, Turkey. Museum of Comparative Zoology, Harvard University, specimen #51621 (Hongo and Meadow 2000).
- Payne** *Sus scrofa*. Measurements based on a sample of modern wild boar, *Sus scrofa libycus*, (male and female; Kizilcahamam, Turkey) from Payne and Bull (1988), Appendix 2.
- Red deer - Cervus Basel** *Cervus elaphus*. Inv.nr. 2271 (male; adult) from Stopp and Deschler-Erb (2018).
- Fallow deer - Dama Haifa** *Dama mesopotamica*. Adult female modern specimen from Israel (id #1047), curated in Archaeozoology Laboratory at the University of Haifa (Harding and Marom 2021).
- Gazelle - Gazella Haifa** *Gazella gazella*. Adult female modern specimen from Israel (id #1037), curated in Archaeozoology Laboratory at the University of Haifa (Harding and Marom 2021).
- Equid - Equus Haifa** *Equus asinus*. Adult male modern specimen from Israel (id #1076), curated in Archaeozoology Laboratory at the University of Haifa (Harding and Marom 2021).
- Johnstone** *Equus caballus*. 3 years old Icelandic mare (all bones fused, female) died in 1961, (Johnstone 2004). Skeleton held at the Zoologische Staatssammlung Munich in Germany. Specimen ID 1961/29.
- European rabbit - Oryctolagus Nottingham** *Oryctolagus cuniculus*. Adult male European rabbit from Audley End, Essex, UK, curated in the reference collection at University of Nottingham Arch department (ID RS139) (Ameen 2021).

The **zoolog** variable referencesDatabase collects all these references. It is structured as a named list of named lists, following the hierarchy described above:

```
str(referencesDatabase, max.level = 2)
```

```
## List of 14
## $ Bos taurus      :List of 3
## ..$ Nieto      :'data.frame': 68 obs. of  4 variables:
## ..$ Basel      :'data.frame': 50 obs. of  4 variables:
## ..$ Johnstone:'data.frame': 24 obs. of  4 variables:
## $ Bos primigenius :List of 1
## ..$ Degerbol:'data.frame': 50 obs. of  4 variables:
## $ Ovis aries      :List of 2
## ..$ Davis      :'data.frame': 23 obs. of  4 variables:
## ..$ Clutton:'data.frame': 71 obs. of  4 variables:
## $ Ovis orientalis :List of 2
## ..$ Basel      :'data.frame': 36 obs. of  4 variables:
## ..$ Uerpmann:'data.frame': 50 obs. of  4 variables:
## $ Capra hircus    :List of 2
## ..$ Basel      :'data.frame': 35 obs. of  4 variables:
## ..$ Clutton:'data.frame': 60 obs. of  4 variables:
## $ Capra aegagrus :List of 1
## ..$ Uerpmann:'data.frame': 50 obs. of  4 variables:
## $ Sus domesticus :List of 1
## ..$ Albarella:'data.frame': 42 obs. of  4 variables:
## $ Sus scrofa      :List of 3
## ..$ Basel:'data.frame': 41 obs. of  4 variables:
## ..$ Hongo:'data.frame': 96 obs. of  4 variables:
## ..$ Payne:'data.frame': 33 obs. of  4 variables:
## $ Cervus elaphus  :List of 1
## ..$ Basel:'data.frame': 14 obs. of  4 variables:
## $ Dama mesopotamica :List of 1
## ..$ Haifa:'data.frame': 60 obs. of  4 variables:
## $ Gazella gazella :List of 1
## ..$ Haifa:'data.frame': 63 obs. of  4 variables:
## $ Equus asinus     :List of 1
## ..$ Haifa:'data.frame': 48 obs. of  4 variables:
## $ Equus caballus  :List of 1
## ..$ Johnstone:'data.frame': 73 obs. of  4 variables:
## $ Oryctolagus cuniculus:List of 1
## ..$ Nottingham:'data.frame': 58 obs. of  4 variables:
```

### Reference Sets

The references' database is organized per taxon. However, in general the zooarchaeological data to be analysed includes several taxa. Thus, the reference dataframe should include one reference standard for each relevant taxon. The **zoolog** variable referenceSets defines four possible references:

referenceSets

	Bos taurus	Bos primigenius	Ovis aries	Ovis orientalis	Capra hircus	Capra aegagrus	Sus
NietoDavisAlbarella	Nieto		Davis				Alba
Basel	Basel			Basel	Basel		
Combi	Nieto		Clutton		Clutton		

Groningen

Degerbol

Uerpmann

Uerpmann

Each row defines a reference set consisting of a reference source for each taxon (column). The function `AssembleReference` allows us to build the reference set taking the selected taxon-specific references from the `referencesDatabase`.

The `zoolog` variable reference is a named list including the references defined by `referenceSets`:

```
str(reference)
```

```
## List of 4
## $ NietoDavisAlbarella:'data.frame': 133 obs. of 4 variables:
## ..$ TAX      : Factor w/ 3 levels "bota","ovar",...: 1 1 1 1 1 1 1 1 1 1 ...
## ..$ EL       : Factor w/ 27 levels "AS","CAL","FE",...: 4 4 4 4 4 4 4 4 4 4 11 ...
## ..$ Measure  : Factor w/ 24 levels "BFd","BFp","BT",...: 8 9 5 7 12 4 3 11 6 8 ...
## ..$ Standard: num [1:133] 259 234 78.3 90.2 29 ...
## $ Basel      : 'data.frame': 176 obs. of 4 variables:
## ..$ TAX      : Factor w/ 5 levels "BOTA","Ovis orientalis",...: 1 1 1 1 1 1 1 1 1 1 ...
## ..$ EL       : Factor w/ 28 levels "Astragalus","Calcaneus",...: 14 14 14 14 5 5 5 13 13 13 ...
## ..$ Measure  : Factor w/ 26 levels "BFd","BFp","BG",...: 21 13 18 3 5 4 19 6 19 5 ...
## ..$ Standard: num [1:176] 65.9 83 66.9 58.1 95.3 ...
## $ Combi      : 'data.frame': 556 obs. of 4 variables:
## ..$ TAX      : Factor w/ 10 levels "bota","OVAR",...: 1 1 1 1 1 1 1 1 1 1 ...
## ..$ EL       : Factor w/ 56 levels "AS","CAL","FE",...: 4 4 4 4 4 4 4 4 4 4 11 ...
## ..$ Measure  : Factor w/ 63 levels "BFd","BFp","BT",...: 8 9 5 7 12 4 3 11 6 8 ...
## ..$ Standard: num [1:556] 259 234 78.3 90.2 29 ...
## $ Groningen  : 'data.frame': 246 obs. of 4 variables:
## ..$ TAX      : Factor w/ 4 levels "Bos primigenius",...: 1 1 1 1 1 1 1 1 1 1 ...
## ..$ EL       : Factor w/ 23 levels "Astragalus","Calcaneus",...: 13 13 13 5 5 5 5 12 12 ...
## ..$ Measure  : Factor w/ 45 levels "BFp","BG","BT",...: 14 12 2 8 9 4 3 13 8 5 ...
## ..$ Standard: num [1:246] 69 70 60 359 309 97 89 46 320 100 ...
```

`reference$Combi` includes the most comprehensive reference for each species so that more measurements can be considered. It is the default reference for computing the [log ratios](#).

If desired, the user can define their own combinations or can also use their own references, which must be a dataframe with the format described above.

## File Structure

`referencesDatabase`, `referenceSets`, and `reference` are exported variables automatically loaded in memory. In addition, `zoolog` provides in the `extdata` folder a set of semicolon separated files (csv), generating them:

`referenceSets.csv` Defines `referenceSets`.

`referencesDatabase.csv` Defines the structure of `referencesDatabase`.

... A csv file for each taxon-specific reference, as named in `referencesDatabase.csv`.

```
utils::read.csv2(system.file("extdata", "referencesDatabase.csv",
                             package = "zoolog"))
```

##	Genus	Taxon	Source
## 1	Cattle - *Bos*	Bos taurus	Nieto
## 2	Cattle - *Bos*	Bos taurus	Basel
## 3	Cattle - *Bos*	Bos taurus	Johnstone
## 4	Cattle - *Bos*	Bos primigenius	Degerbol
## 5	Sheep - *Ovis*	Ovis aries	Davis
## 6	Sheep - *Ovis*	Ovis aries	Clutton
## 7	Sheep - *Ovis*	Ovis orientalis	Basel
## 8	Sheep - *Ovis*	Ovis orientalis	Uerpmann
## 9	Goat - *Capra*	Capra hircus	Basel
## 10	Goat - *Capra*	Capra hircus	Clutton
## 11	Goat - *Capra*	Capra aegagrus	Uerpmann
## 12	Pig - *Sus*	Sus domesticus	Albarella
## 13	Pig - *Sus*	Sus scrofa	Basel
## 14	Pig - *Sus*	Sus scrofa	Hongo
## 15	Pig - *Sus*	Sus scrofa	Payne
## 16	Red deer - *Cervus*	Cervus elaphus	Basel
## 17	Fallow deer - *Dama*	Dama mesopotamica	Haifa
## 18	Gazelle - *Gazella*	Gazella gazella	Haifa
## 19	Equid - *Equus*	Equus asinus	Haifa
## 20	Equid - *Equus*	Equus caballus	Johnstone
## 21	European rabbit - *Oryctolagus*	Oryctolagus cuniculus	Nottingham
##	Filename		
## 1	referenceCattle_Nieto.csv		
## 2	referenceCattle_Basel.csv		
## 3	referenceCattle_Johnstone.csv		
## 4	referenceCattle_Degerbol.csv		
## 5	referenceSheep_Davis.csv		
## 6	referenceSheep_Clutton.csv		
## 7	referenceSheep_Basel.csv		
## 8	referenceSheep_Uerpmann.csv		
## 9	referenceGoat_Basel.csv		
## 10	referenceGoat_Clutton.csv		
## 11	referenceGoat_Uerpmann.csv		
## 12	referencePig_Albarella.csv		
## 13	referencePig_Basel.csv		
## 14	referencePig_Hongo.csv		
## 15	referencePig_Payne.csv		
## 16	referenceRedDeer_Basel.csv		
## 17	referenceDama_Haifa.csv		
## 18	referenceGazelle_Haifa.csv		
## 19	referenceEquid_Haifa.csv		
## 20	referenceEquid_Johnstone.csv		
## 21	referenceRabbit_Nottingham.csv		

### Acknowledgement

We are grateful to Barbara Stopp and Sabine Deschler-Erb for providing the Basel references for cattle, sheep, goat, wild boar, and red deer (Stopp and Deschler-Erb 2018), together with the permission to publish them as part of **zoolog**.

We thank also Francesca Slim and Dimitris Filioglou for providing the references for aurochs, mouflon, wild goat, and wild boar (Degerbøl and Fredskild 1970; Uerpmann and Uerpmann 1994; Hongo and Meadow 2000) in the Groningen set.

We thank Claudia Minniti for providing Johnstone's reference for cattle. (Johnstone and Albarella 2002).

We are also grateful to Sierra Harding and Nimrod Marom for providing the Haifa standard measurements for donkey, gazelle, and fallow deer (Harding and Marom 2021).

We thank Carly Ameen for providing references for horse (Johnstone 2004) and rabbit (Ameen 2021).

### References

Albarella U, Payne S (2005). "Neolithic pigs from Durrington Walls, Wiltshire, England: a biometrical database." *Journal of Archaeological Science*, **32**(4), 589–599.

Ameen C (2021). "Measurements from an adult male specimen from Audley End, Essex, UK. in the reference collection at the University of Nottingham Archaeology Department under ID RS139." Personal communication, included permission to publish them as part of the package **zoolog**.

Clutton-Brock J, Dennis-Bryan K, Armitage PL, Jewell PA (1990). "Osteology of the Soay sheep." *Bulletin of the British Museum, Natural History. Zoology*, **56**(1), 1–56.

Davis SJ (1996). "Measurements of a group of adult female Shetland sheep skeletons from a single flock: a baseline for zooarchaeologists." *Journal of archaeological science*, **23**(4), 593–612.

Degerbøl M, Fredskild B (1970). *The Urus (Bos Primigenius Bojanus) and Neolithic Domesticated Cattle (Bos Taurus Domesticus Linné) in Denmark: Zoological and Palynological Investigations*, Biologiske skrifter, 17:1. København, (Munksgaard).

Harding S, Marom N (2021). "Measurements compiled for the Zooarchaeology of Southern Phoenicia (ZSP) Project, from the reference collection in the Leon Recanati Institute for Maritime Studies (RIMS, Department of Maritime Civilizations, University of Haifa, Israel)." Personal communication, included permission to publish them as part of the package **zoolog**.

Hongo H, Meadow RH (2000). "Faunal remains from Prepottery Neolithic levels at Çayönü, south-eastern Turkey: a preliminary report focusing on pigs (*Sus* sp.)." In *Archaeozoology of the Near East IVA Proceedings of the fourth international symposium on the archaeozoology of southwestern Asia and adjacent areas*. Groningen: ARC Publications, 121–139.

Johnstone C, Albarella U (2002). "The Late Iron Age and Romano-British Mammal and Bird Bone Assemblage from Elms Farm, Heybridge, Essex (Site Code: Hyef93-95)." Technical Report Report 45/2002, tab.16, p. 70, Centre for Archaeology.

Johnstone CJ (2004). *A biometric study of equids in the Roman world*. Ph.D. thesis, University of York.

Nieto-Espinet A (2018). “Element measure standard biometrical data from a cow dated to the Early Bronze Age (Minferri, Catalonia).” doi: [10.13140/RG.2.2.13512.78081](https://doi.org/10.13140/RG.2.2.13512.78081).

Payne S, Bull G (1988). “Components of variation in measurements of pig bones and teeth, and the use of measurements to distinguish wild from domestic pig remains.” *Archaeozoologia*, **2**(1), 27–66.

Stopp B, Deschler-Erb S (2018). “Measurements compiled from the reference collection in the Integrative Prähistorische und Naturwissenschaftliche Archäologie (IPNA, University of Basel, Switzerland).” Personal communication, included permission to publish them as part of the package *zoolog*.

Uerpmann M, Uerpmann H (1994). “Animal bone finds from excavation 520 at Qala’at al-Bahrain.” In Hojlund F, Andersen HH (eds.), *Gala’at Al-Bahrain. 1 The Northern City Wall And The Islamic Fortress*, 417–444. Jutland Archaeological Society.

---

RemoveNACases

*Remove Cases Missing All Measurements*

---

## Description

Function to remove the table rows for which all measurements of interest are non-available (NA). A particular list of measurement names can be explicitly provided or selected by a common initial pattern. The default setting removes the rows with no log-ratio available.

## Usage

```
RemoveNACases(data, measureNames = NULL, prefix = logPrefix)
```

## Arguments

<code>data</code>	A dataframe with the input measurements.
<code>measureNames</code>	A vector of characters with the list of measurements to be considered for missing values. If NULL (default), all measurements starting by <code>prefix</code> are considered.
<code>prefix</code>	A character string with the initial pattern to select the list of measurements. The default is given by the internal variable <code>logPrefix</code> . It is in effect only when <code>measureNames = NULL</code> .

## Value

A dataframe with the same columns as the input dataframe but removing the rows with missing values for all measurements in the list.



**Examples**

```

## Read an example dataset:
dataFile <- system.file("extdata", "dataValenzuelaLamas2008.csv.gz",
                        package = "zoolog")
dataExample <- utils::read.csv2(dataFile,
                                na.strings = "",
                                encoding = "UTF-8",
                                stringsAsFactors = TRUE)

## We can observe the first lines (excluding some columns for visibility):
head(dataExample)[, -c(6:20,32:64)]

## Remove the cases not including any measurement present in the reference.
refMeasureNames <- levels(factor(reference$Combi$Measure))
refMeasureNames
dataExamplePruned <- RemoveNACases(dataExample,
                                   measureNames = refMeasureNames)

## The first lines of the output data frame show at least one available
## measurement value in the selected list:
head(dataExamplePruned)[, -c(6:20,32:64)]

## If we compute first the log-ratios
dataExampleWithLogs <- LogRatios(dataExample)
## the cases not including any log-ratio can be removed with the
## default logPrefix
dataExampleWithLogsPruned <- RemoveNACases(dataExampleWithLogs)
head(dataExampleWithLogsPruned)[, -c(6:20,32:64)]

```

---

StandardizeNomenclature

*Standardize Nomenclature*


---

**Description**

Functions to map the user provided nomenclature into a standard one as defined in a thesaurus.

**Usage**

```
StandardizeNomenclature(x, thesaurus, mark.unknown = FALSE)
```

```
StandardizeDataSet(data, thesaurusSet = zoologThesaurus)
```

**Arguments**

x	Character vector.
thesaurus	A thesaurus object.
mark.unknown	Logical. If FALSE (default) the strings not found in the thesaurus are kept without change. If TRUE the strings not in the thesaurus are set to NA.
data	A data frame.
thesaurusSet	A thesaurus set.

**Details**

StandardizeNomenclature standardizes a character vector according to a given thesaurus.

StandardizeDataSet standardizes column names and values of a data frame according to a thesaurus set.

**Value**

StandardizeNomenclature returns a vector of the same length as the input vector *x*. The names present in the thesaurus are set to their corresponding category. The names not in the thesaurus are kept unchanged if `mark.unknown=FALSE` (default) and set to NA if `mark.unknown=TRUE`.

StandardizeDataSet returns a data frame with the same structure as the input data, but standardizing its nomenclature according to a thesaurus set including appropriate thesauri for its column names and for the values of a set of columns.

**See Also**

[zoologThesaurus](#) for a description of the thesaurus and thesaurus set structure, [ThesaurusReaderWriter](#), [ThesaurusManagement](#)

**Examples**

```
## Select the thesaurus for taxa present in the thesaurus set
## zoolog::zoologThesaurus:
thesaurus <- zoologThesaurus$taxon
thesaurus
## Standardize an heterodox vector of taxa:
StandardizeNomenclature(c("bota", "rabbit", "pig", "cattle"),
                        thesaurus)
## Observe that "rabbit" is kept unchanged since it is not included in
## any thesaurus category.
## But if mark.unknown is set to TRUE, it is marked as NA:
StandardizeNomenclature(c("bota", "rabbit", "pig", "cattle"),
                        thesaurus, mark.unknown = TRUE)

## This thesaurus is not case sensitive:
attr(thesaurus, "caseSensitive") # == FALSE
## Thus, names are recognized independently of their case:
StandardizeNomenclature(c("bota", "BOTA", "Bota", "boTa"),
                        thesaurus)

## Load an example data frame:
dataFile <- system.file("extdata", "dataValenzuelaLamas2008.csv.gz",
                        package = "zoolog")
dataExample <- utils::read.csv2(dataFile,
                                na.strings = "",
                                encoding = "UTF-8",
                                stringsAsFactors = TRUE)

## Observe mainly the first columns:
head(dataExample[,1:5])
## Standardize the dataset:
```

```
dataStandardized <- StandardizeDataSet(dataExample, zoologThesaurus)
head(dataStandardized[,1:5])
```

---

ThesaurusManagement    *Thesaurus Management*

---

## Description

Functions to modify and check thesauri.

## Usage

```
NewThesaurus(
  caseSensitive = FALSE,
  accentSensitive = FALSE,
  punctuationSensitive = FALSE
)
```

```
AddToThesaurus(thesaurus, newName, category)
```

```
RemoveRepeatedNames(thesaurus)
```

```
ThesaurusAmbiguity(thesaurus)
```

## Arguments

<code>caseSensitive</code>	Logical. They set the case, accent, and punctuation sensitivity (FALSE by default) of the thesaurus.
<code>accentSensitive</code>	Logical. They set the case, accent, and punctuation sensitivity (FALSE by default) of the thesaurus.
<code>punctuationSensitive</code>	Logical. They set the case, accent, and punctuation sensitivity (FALSE by default) of the thesaurus.
<code>thesaurus</code>	A thesaurus object.
<code>newName</code>	Character vector with new names to be added to the thesaurus.
<code>category</code>	Character vector identifying the classes where the new names should be included.

## Value

`NewThesaurus` returns an empty thesaurus. This can then be populated by `AddToThesaurus`.

`AddToThesaurus` returns the input thesaurus complemented with new names in the categories identified. If any of the categories is not present in the input thesaurus, new categories are added as required.

RemoveRepeatedNames returns the input thesaurus pruned of redundant names in each category. The redundancy is evaluated in agreement with the case and accent sensitivity of the thesaurus.

ThesaurusAmbiguity returns FALSE if no ambiguity is present. When any ambiguity is found, it returns TRUE with an attribute errmessage including the names present in more than one category and the the involved categories. This is internally used by [ReadThesaurus](#) and [AddToThesaurus](#) to generate an error in case they attempt to read or generate an ambiguous thesaurus.

### See Also

[zoologThesaurus](#) for a description of the thesaurus and thesaurus set structure,  
[ReadThesaurus](#), [WriteThesaurus](#), [StandardizeNomenclature](#)

### Examples

```
## Load an example thesaurus:
thesaurus <- ReadThesaurus(system.file("extdata", "taxonThesaurus.csv",
                                     package="zoolog"))

## with categories
names(thesaurus) # "bos taurus" "ovis aries" "sus domesticus"
## Add names to several categories:
thesaurusExtended <- AddToThesaurus(thesaurus,
                                   c("Kuh", "Schwein"),
                                   c("bos taurus","sus domesticus"))

## This adds the name "Kuh" to the category "bos taurus" and
## the name "Schwein" to the category "sus domesticus".

## Generate a new thesaurus and populate it with two categories
## ("red" and "blue"):
thesaurusNew <- NewThesaurus()
thesaurusNew <- AddToThesaurus(thesaurusNew,
                              c("scarlet", "vermilion", "ruby", "cherry",
                                "carmine", "wine"),
                              "red")

thesaurusNew
thesaurusNew <- AddToThesaurus(thesaurusNew,
                              c("sky blue", "azure", "sapphire", "cerulean",
                                "navy", "lapis lazuli", "indigo", "cyan"),
                              "blue")

thesaurusNew

## Attempt to generate an ambiguous thesaurus
try(AddToThesaurus(thesaurusNew, "scarlet", "blue"))

## Remove repeated names in the same category:
thesaurusWithRepetitions <- AddToThesaurus(thesaurusNew,
                                           c("scarlet", "ruby"), "red")

thesaurusWithRepetitions
RemoveRepeatedNames(thesaurusWithRepetitions)
```

**Description**

Functions to read and write thesauri and thesaurus sets.

**Usage**

```
ReadThesaurus(  
    file,  
    caseSensitive = FALSE,  
    accentSensitive = FALSE,  
    punctuationSensitive = FALSE  
)  
  
ReadThesaurusSet(file)  
  
WriteThesaurus(thesaurus, file)  
  
WriteThesaurusSet(thesaurusSet, file)
```

**Arguments**

<code>file</code>	Name of a file.
<code>caseSensitive</code> , <code>accentSensitive</code> , <code>punctuationSensitive</code>	Logical. They set the case, accent, and punctuation sensitivity (FALSE by default) of the thesaurus.
<code>thesaurus</code>	A thesaurus object.
<code>thesaurusSet</code>	A thesaurus set.

**Value**

`WriteThesaurus` and `WriteThesaurusSet` create or overwrite the corresponding files. No value is returned.

`ReadThesaurus` and `ReadThesaurusSet` return the read thesaurus or thesaurusSet, respectively.

**See Also**

[zoologThesaurus](#) for a description of the thesaurus and thesaurus set structure,  
[ThesaurusManagement](#), [StandardizeNomenclature](#)

## Examples

```
## Read a thesaurus for taxa:
thesaurusFile <- system.file("extdata", "taxonThesaurus.csv", package="zoolog")
thesaurus <- ReadThesaurus(thesaurusFile)
## The attributes of the thesaurus include the fields 'caseSensitive',
## 'accentSensitive', and 'punctuationSensitive', all FALSE by default.
attributes(thesaurus)

## Any of them can be set by the user if desired:
thesaurus2 <- ReadThesaurus(thesaurusFile, accentSensitive = TRUE)
attributes(thesaurus2)

## Write the thesaurus to a file:
fileExample <- file.path(tempdir(), "thesaurusExample.csv")
WriteThesaurus(thesaurus, fileExample)
## Replace tempdir() for your preferred local path if you want to easily
## examine the written file.

## Read a thesaurus set:
thesaurusSetFile <- system.file("extdata", "zoologThesaurusSet.csv", package="zoolog")
thesaurusSet <- ReadThesaurusSet(thesaurusSetFile)
## The attributes of the thesaurus set include information of the constituent
## thesauri: names, source file names, and their mode of application on datasets.
attributes(thesaurusSet)
## The attributes of each thesaurus are also set by 'ReadThesaurusSet'.
attributes(thesaurusSet$measure)

## Write the thesaurus set to a file:
fileSetExample <- file.path(tempdir(), "thesaurusSetExample.csv")
WriteThesaurusSet(thesaurusSet, fileSetExample)
## It writes the thesaurus-set main data frame and each of the included
## thesaurus files.
## Again, replace tempdir() for your preferred local path if you want to
## easily examine the written files.
```

---

zoologThesaurus

*Thesaurus Set for zoolog*

---

## Description

The thesaurus set defined for the package **zoolog**. This is used to make the methods robust to different nomenclatures used in datasets created by different authors. The user can also use other thesaurus sets, or can modify the provided thesaurus set (see [ThesaurusManagement](#) and [ThesaurusReaderWriter](#)).

## Usage

```
zoologThesaurus
```

## Format

A thesaurus set is a list of thesauri with additional attributes:

**names** Character vector with the name of each thesaurus.

**applyToColNames** Logical vector indicating whether each thesaurus should be applied to the column names of the data frame.

**applyToColValues** Logical vector indicating whether each thesaurus should be applied to the values in the corresponding column of the data frame.

**filename** Character vector with the source file of each thesaurus.

The examples below show the list of four thesauri included in the provided zoologThesaurus.

Each thesaurus is a data frame also with additional attributes. Each column of the data frame is a category of names with equivalent meaning in the intended application. The column name identifies the category and is used as the standard when applying [StandardizeNomenclature](#).

The names in each column (category) must not be included in any other column, since this would make the thesaurus ambiguous (see [ThesaurusAmbiguity](#)).

Each thesaurus has the following attributes:

**names** The standard name for the categories.

**class** "data.frame"

**row.names** Irrelevant

**caseSensitive** Logical indicating whether the names in the thesaurus should be considered case-sensitive.

**accentSensitive** Logical indicating whether the names in the thesaurus should be differentiated by the presence of accent marks.

**punctuationSensitive** Logical indicating whether the names in the thesaurus should be differentiated by the presence of punctuation marks.

The examples below show the content and characteristics of the first thesaurus in zoologThesaurus.

## File Structure

zoologThesaurus is an exported variable automatically loaded in memory. In addition, the source files generating it are included in the **zoolog** extdata folder. There is one file for the thesaurus set main structure and one file for each included thesaurus. All of them are in semicolon separated format. Thus, they can be examined in any text editor or imported into any spreadsheet application. The files are:

zoologThesaurusSet.csv Defines the main structure of the thesaurus set. It has a row for each thesaurus and seven columns (*ThesaurusName*, *FileName*, *CaseSensitive*, *AccentSensitive*, *PunctuationSensitive*, *ApplyToColNames*, and *ApplyToColValues*). Their meaning coincides with the description above. Observe that the case, accent, and punctuation sensitiveness is stored here, instead of in each thesaurus.

identifierThesaurus.csv Thesaurus for the identifiers used in [LogRatios](#) to identify the bone types and the measure names in the data and the references. It has for columns: *Taxon*, *Element*, *Measure*, and *Standard*.

taxonThesaurus.csv Thesaurus for the taxa. There is one column for each category of taxon considered.

elementThesaurus.csv Thesaurus for the skeletal elements. One column for each category.

measureThesaurus.csv Thesaurus for the measure names. One column for each category.

### **Examples**

```
## List of thesaurus names and characteristics in the thesaurus set:  
attributes(zoologThesaurus)  
## Content of the first thesaurus:  
zoologThesaurus$identifier  
attributes(zoologThesaurus$identifier)
```



# Index

- \* **datasets**
  - referencesDatabase, [10](#)
  - zoologThesaurus, [22](#)
- [%in%](#), [6](#), [7](#)
- AddToThesaurus, [20](#)
- AddToThesaurus (ThesaurusManagement), [19](#)
- AssembleReference, [2](#), [13](#)
- CondenseLogs, [3](#)
- dataValenzuelaLamas2008, [5](#)
- InCategory, [6](#)
- is.element, [6](#)
- log ratios, [13](#)
- LogRatios, [7](#), [23](#)
- NewThesaurus (ThesaurusManagement), [19](#)
- ReadThesaurus, [20](#)
- ReadThesaurus (ThesaurusReaderWriter), [21](#)
- ReadThesaurusSet (ThesaurusReaderWriter), [21](#)
- reference (referencesDatabase), [10](#)
- referencesDatabase, [2](#), [10](#)
- referenceSets (referencesDatabase), [10](#)
- RemoveNACases, [16](#)
- RemoveRepeatedNames (ThesaurusManagement), [19](#)
- StandardizeDataSet (StandardizeNomenclature), [17](#)
- StandardizeNomenclature, [17](#), [20](#), [21](#), [23](#)
- ThesaurusAmbiguity, [23](#)
- ThesaurusAmbiguity (ThesaurusManagement), [19](#)
- ThesaurusManagement, [18](#), [19](#), [21](#), [22](#)
- ThesaurusReaderWriter, [18](#), [21](#), [22](#)
- WriteThesaurus, [20](#)
- WriteThesaurus (ThesaurusReaderWriter), [21](#)
- WriteThesaurusSet (ThesaurusReaderWriter), [21](#)
- zoologThesaurus, [7](#), [18](#), [20](#), [21](#), [22](#)