

# Package ‘TEKRABber’

July 3, 2023

**Type** Package

**Title** An R package estimates the correlations of orthologs and transposable elements between two species

**Version** 1.4.0

**Description** TEKRABber is made to provide a user-friendly pipeline for comparing orthologs and transposable elements (TEs) between two species. It considers the orthology confidence between two species from BioMart to normalize expression counts and detect differentially expressed orthologs/TEs. Then it provides one to one correlation analysis for desired orthologs and TEs. There is also an app function to have a first insight on the result. Users can prepare orthologs/TEs RNA-seq expression data by their own preference to run TEKRABber following the data structure mentioned in the vignettes.

**URL** <https://github.com/ferygood/TEKRABber>

**BugReports** <https://github.com/ferygood/TEKRABber/issues>

**Encoding** UTF-8

**License** GPL (>= 2)

**Imports** apeglm, biomaRt, dplyr, DESeq2, magrittr, Rcpp (>= 1.0.7), SCBN, SummarizedExperiment, stats, utils

**LinkingTo** Rcpp

**Depends** R (>= 4.1)

**LazyData** false

**Suggests** BiocStyle, ggpubr, rmarkdown, shiny, knitr, testthat (>= 3.0.0)

**VignetteBuilder** knitr

**VignetteEngine** knitr

**RoxygenNote** 7.1.2

**biocViews** DifferentialExpression, Normalization, Transcription, GeneExpression

**git\_url** <https://git.bioconductor.org/packages/TEKRABber>

**git\_branch** RELEASE\_3\_17

**git\_last\_commit** d5e1844

**git\_last\_commit\_date** 2023-04-25

**Date/Publication** 2023-07-02

**Author** Yao-Chung Chen [aut, cre] (<<https://orcid.org/0000-0002-9927-9130>>),  
Katja Nowick [aut] (<<https://orcid.org/0000-0003-3993-4479>>)

**Maintainer** Yao-Chung Chen <yao-chung.chen@fu-berlin.de>

## R topics documented:

appTEKRABber . . . . .	2
assay_tekcorrset . . . . .	4
corrOrthologTE . . . . .	4
ctCorr . . . . .	5
ctInputDE . . . . .	6
DECorrInputs . . . . .	7
DEgeneTE . . . . .	8
fetchDataHmChimp . . . . .	9
hg38_panTro6_rmsk . . . . .	9
orthologScale . . . . .	10
rpp_corr . . . . .	11
speciesCorr . . . . .	12
speciesCounts . . . . .	12
TEKRABber . . . . .	13
<b>Index</b>	<b>14</b>

---

appTEKRABber

*Visualize TEKRABber results with shiny app*

---

### Description

To help user explore their results using TEKRABber, this function visualizes the results using a self-written shiny app with two tabs, including the expression and correlation of genes and TEs. To run it, you need to create four variables and assign them with your DE result, correlation results and metadata to appDE, appRef, appCompare and appMeta. Please see the example below for more details.

### Usage

```
appTEKRABber()
```

### Value

An app to display differentially expressed genes/TEs and the correlation results

**Examples**

```

data(fetchDataHmChimp)
fetchData <- fetchDataHmChimp

inputBundle <- DECorrInputs(fetchData)

# create metadata for DE analysis
meta <- data.frame(species=c(rep("human", ncol(fetchData$geneRef) - 1),
  rep("chimpanzee", ncol(fetchData$geneCompare) - 1))
)
rownames(meta) <- colnames(inputBundle$geneInputDESeq2)
meta$species <- factor(meta$species, levels = c("human", "chimpanzee"))

# DE analysis
hmchimpDE <- DEgeneTE(
  geneTable = inputBundle$geneInputDESeq2,
  teTable = inputBundle$teInputDESeq2,
  metadata = meta,
  expDesign = TRUE
)

data(speciesCorr)
hmGeneCorrInput <- assay_tekcorrset(speciesCorr, "gene", "human")
hmTECorrInput <- assay_tekcorrset(speciesCorr, "te", "human")
chimpGeneCorrInput <- assay_tekcorrset(speciesCorr, "gene", "chimpanzee")
chimpTECorrInput <- assay_tekcorrset(speciesCorr, "te", "chimpanzee")

# Correlation analysis
hmCorrResult <- corrOrthologTE(
  geneInput = hmGeneCorrInput,
  teInput = hmTECorrInput,
  corrMethod = "pearson",
  padjMethod = "fdr"
)
chimpCorrResult <- corrOrthologTE(
  geneInput = chimpGeneCorrInput,
  teInput = chimpTECorrInput,
  corrMethod = "pearson",
  padjMethod = "fdr"
)

# assign results and metadata to appDE, appRef, appCompare, and appMeta
appDE <- hmchimpDE
appRef <- hmCorrResult
appCompare <- chimpCorrResult
appMeta <- meta

if (interactive()){
  appTEKRABber()
}

```

---

assay_tekcorrset	<i>Access genes and transposable elements expression data</i>
------------------	---

---

### Description

a function only used for accessing the expression data from a TekCorrSet class object to demonstrate examples in vignettes. demonstration.

### Usage

```
assay_tekcorrset(tecorrset, expType, sample)
```

### Arguments

tecorrset	TekCorrSet object
expType	Indicate which data you want to access. It should be "gene" or "te".
sample	The species name or experimental design. It should be "human" and "chimpanzee" when you are running comparing species design. "control" and "treatment" are for running same species design.

### Value

a dataframe contains expression genes or transposable elements.

### Examples

```
data(speciesCorr)
hmGeneCorrInput <- assay_tekcorrset(speciesCorr, "gene", "human")
hmTECorrInput <- assay_tekcorrset(speciesCorr, "te", "human")
chimpGeneCorrInput <- assay_tekcorrset(speciesCorr, "gene", "chimpanzee")
chimpTECorrInput <- assay_tekcorrset(speciesCorr, "te", "chimpanzee")
```

---

corrOrthologTE	<i>Estimate correlation comparing orthologs and TEs</i>
----------------	---

---

### Description

To estimate correlation comparing orthologs and TEs one-by-one from inputs. You can specify the correlation and adjusted p-value methods (see details in parameters). If you want to save your outputs instead of just returning them, please specify the fileDir and fileName with the extension .csv. The default fileName is TEKRABber\_geneTECorrReusult.csv.

**Usage**

```
corrOrthologTE(geneInput, teInput, corrMethod = "pearson",
  padjMethod = "fdr", fileDir=NULL, fileName="TEKRABber_geneTECorrResult.csv")
```

**Arguments**

geneInput	gene count input for correlation from using DECorrInputs()
teInput	te count input for correlation from using DECorrInputs()
corrMethod	correlation method, including pearson, kendall, spearman. Default is pearson.
padjMethod	method to return adjusted p-value, and default is fdr. See ?p.adjust
fileDir	the name of directory for saving output files. Default is NULL.
fileName	the name for saving output files. Default is "TEKRABber_geneTECorrResult.csv"

**Value**

a dataframe includes correlation coefficient, pvalue, padj

**Examples**

```
library(SummarizedExperiment)
data(speciesCorr)
hmGeneCorrInput <- assay_tekcorrset(speciesCorr, "gene", "human")
hmTECorrInput <- assay_tekcorrset(speciesCorr, "te", "human")

corrOrthologTE(
  geneInput=hmGeneCorrInput,
  teInput=hmTECorrInput,
  corrMethod="pearson",
  padjMethod="fdr",
  fileDir=NULL
)
```

---

ctCorr	<i>Normalized Gene/TE expression toy data in control and treatment in same species for correlation analysis</i>
--------	---

---

**Description**

Dataset contains gene/TE expression data from control and treatment based on syn8466812 RNA-seq (Allen M et al., 2016) for correlation analysis. These data were also modified due to confidential agreement. Therefore, it cannot represent the original data. For a quick demonstration, we only use 10 genes and 10 transposable elements.

**Usage**

```
data(ctCorr)
```

**Format**

An object of class "TekCorrSet" which contains 4 expression counts and you can access it specifying the parameters using `assay_tekcorrset()`:

```
assay_tekcorrset(ctCorr, "gene", "control") control gene expression data
assay_tekcorrset(ctCorr, "te", "control") control TE expression data
assay_tekcorrset(ctCorr, "gene", "treatment") treatment gene expression data
assay_tekcorrset(ctCorr, "te", "treatment") treatment TE expression data
```

**Examples**

```
data(ctCorr)
geneConCorrInput <- assay_tekcorrset(ctCorr, "gene", "control")
teConCorrInput <- assay_tekcorrset(ctCorr, "te", "control")
geneTreatCorrInput <- assay_tekcorrset(ctCorr, "gene", "treatment")
teTreatCorrInput <- assay_tekcorrset(ctCorr, "te", "treatment")
```

---

ctInputDE

*Input expression data of gene/TE for differentially expressed analysis within same species*

---

**Description**

TEKRABber can also be used comparing orthologs and transposable elements within same species, i.e., control and treatment. Here we provide an example data for demonstration. This data was based on syn8466812 RNA-seq (Allen M et al., 2016). However, the expression data was modified due to confidential agreement. Therefore, it cannot represent the original data.

**Usage**

```
data(ctInputDE)
```

**Format**

An object contains 2 expression data:

```
gene input gene data for DE analysis comparing control and treatment
te input TE data for DE analysis comparing control and treatment
```

**Examples**

```
data(ctInputDE)
geneInputDE <- ctInputDE$gene
teInputDE <- ctInputDE$te
```

---

**DECORRInputs***Generate all the input files for TEKRABber downstream analysis*

---

**Description**

Generate all the inputs files for differentially expressed orthologous genes/TEs analysis, and for correlation analysis. The output is a list containing 6 dataframes.

**Usage**

```
DECORRInputs(fetchData)
```

**Arguments**

```
fetchData      output list from TEKRABber::orthologScale()
```

**Value**

create inputs for DE analysis and correlations: (1) geneInputDESeq2 (2) teInputDESeq2 (3) geneCorrInputRef (4) geneCorrInputCompare (5) TECorrInputRef (6) TECorrInputCompare

**Examples**

```
data(speciesCounts)
data(hg38_panTro6_rmsk)
hmGene <- speciesCounts$hmGene
chimpGene <- speciesCounts$chimpGene
hmTE <- speciesCounts$hmTE
chimpTE <- speciesCounts$chimpTE

## For demonstration, here we only select 1000 rows to save time
set.seed(1234)
hmGeneSample <- hmGene[sample(nrow(hmGene), 1000), ]
chimpGeneSample <- chimpGene[sample(nrow(chimpGene), 1000), ]

fetchData <- orthologScale(
  speciesRef = "hsapiens",
  speciesCompare = "ptroglodytes",
  geneCountRef = hmGeneSample,
  geneCountCompare = chimpGeneSample,
  teCountRef = hmTE,
  teCountCompare = chimpTE,
  rmsk = hg38_panTro6_rmsk
)

inputBundle <- DECORRInputs(fetchData)
```

DEgeneTE

*Estimate differentially expressed genes and TEs***Description**

To estimate differentially expressed genes and TEs, DEgeneTE() takes gene inputs and TE inputs from the results using the DECorrInputs function. You need to specify your metadata and expDesign based on your design. If you also want to save the output, please specify the fileDir parameter.

**Usage**

```
DEgeneTE(geneTable, teTable, metadata, expDesign=TRUE, fileDir=NULL)
```

**Arguments**

geneTable	gene input table from using DECorrInputs()
teTable	TE input table from using DECorrInputs()
metadata	an one column dataframe with rownames same as the column name of gene/te count table. Column name must be <b>species</b> or <b>experiment</b> .
expDesign	Logic value for comparing between or within species. <b>TRUE</b> for comparing between two species, and <b>FALSE</b> for comparing between control and treatment.
fileDir	the name and path of directory for saving output files. Default is NULL.

**Value**

return DESeq2 res and normalized gene counts.

**Examples**

```
## comparing between species:
## (1) set expDesign = TRUE
## (2) column name of metadata needs to be "species".

data(fetchDataHmChimp)
fetchData <- fetchDataHmChimp

inputBundle <- DECorrInputs(fetchData)

meta <- data.frame(species=c(rep("human", ncol(fetchData$geneRef) - 1),
  rep("chimpanzee", ncol(fetchData$geneCompare) - 1))
)
rownames(meta) <- colnames(inputBundle$geneInputDESeq2)
meta$species <- factor(meta$species, levels = c("human", "chimpanzee"))

hmchimpDE <- DEgeneTE(
  geneTable = inputBundle$geneInputDESeq2,
  teTable = inputBundle$teInputDESeq2,
  metadata = meta,
```



```

    expDesign = TRUE
  )

```

---

fetchDataHmChimp	<i>Example output comparing human and chimpanzee data using orthologScale()</i>
------------------	---

---

### Description

An output list of data contains 7 elements after using orthologScale(), including (1) orthology table comparing human and chimpanzee. (2) scaling factor for orthologous genes (3) gene count table from reference species (4) gene count table from species you want to compare (5) scaling factor for TEs (6) TE count table from reference species (7) TE count table from the species you want to compare. The aim to provide this dataset is to save time for user running the vignettes and give a template for demonstration.

### Usage

```
data(fetchDataHmChimp)
```

### Format

An object contains 2 elements:

**orthologTable** orthology information from Ensembl  
**scaleFactor** scaling factor to normalize data

### Examples

```

data(fetchDataHmChimp)
fetchData <- fetchDataHmChimp
fetchData$orthologTable
fetchData$scaleFactor

```

---

hg38_panTro6_rmsk	<i>Repeatmasker track annotations with human and chimpanzee</i>
-------------------	---

---

### Description

This Repeatmasker track annotations table was first downloaded from UCSC Genome Table Browser and it included the name, class, and average gene length in repeats(transposable elements). This data is used for demonstrate an example for user how to provide a annotation table to normalize their data which in this case comparing human(hg38) to chimpanzee(panTro6).

### Usage

```
data(hg38_panTro6_rmsk)
```

**Format**

An object of class `grouped_df` (inherits from `tbl_df`, `tbl`, `data.frame`) with 12550 rows and 4 columns.

**Examples**

```
data(hg38_panTro6_rmsk)
```

---

orthologScale	<i>Normalized orthologous genes and TEs between two species</i>
---------------	---

---

**Description**

Normalize orthologous genes and TEs between two species with a scaling factor using their expression level and gene lengths.

**Usage**

```
orthologScale(speciesRef, speciesCompare, geneCountRef,
geneCountCompare, teCountRef, teCountCompare, rmsk)
```

**Arguments**

speciesRef	The scientific name for your reference species. i.e., <code>hsapiens</code>
speciesCompare	The scientific name for your species to compare. i.e., <code>ptroglodytes</code>
geneCountRef	Gene count from your reference species. First column should be Ensembl gene ID.
geneCountCompare	Gene count from the species you want to compare. First column should be Ensembl gene ID.
teCountRef	TE count from your reference species. First column should be <code>teName</code> .
teCountCompare	TE count from the species you want to compare. First column should be <code>teName</code> .
rmsk	a repeatmasker table including 4 columns: (1) the name of TE (2) the class of TE (3) The average length of that TE from your reference species (4) The average length of that TE from the species you want to compare.

**Value**

a list of outputs: (1) `orthologTable`, orthology information (2) `c_ortholog`, scaling factor for orthologous genes (3) `geneRef`, gene count table for reference species (4) `geneCompare`, normalized gene count table for species compared (5) `c_te`, scaling factor for TEs (6) `teRef`, TE count table for reference species (7) `teCompare`, normalized TE count table for species compared.

**Examples**

```
data(speciesCounts)
data(hg38_panTro6_rmsk)
hmGene <- speciesCounts$hmGene
chimpGene <- speciesCounts$chimpGene
hmTE <- speciesCounts$hmTE
chimpTE <- speciesCounts$chimpTE

## For demonstration, here we only select 1000 rows to save time
set.seed(1234)
hmGeneSample <- hmGene[sample(nrow(hmGene), 1000), ]
chimpGeneSample <- chimpGene[sample(nrow(chimpGene), 1000), ]

fetchData <- orthologScale(
  speciesRef = "hsapiens",
  speciesCompare = "ptroglodytes",
  geneCountRef = hmGeneSample,
  geneCountCompare = chimpGeneSample,
  teCountRef = hmTE,
  teCountCompare = chimpTE,
  rmsk = hg38_panTro6_rmsk
)
```

---

rcpp\_corr

*Estimate the correlation between genes and transposable elements*

---

**Description**

Estimate the correlation between genes and transposable elements

**Usage**

```
rcpp_corr(df1, df2, Method)
```

**Arguments**

df1	First dataframe
df2	Second dataframe
Method	correlation method

**Value**

a dataframe containing correlation results

---

speciesCorr	<i>A subsets of normalized Gene/TE expression data from human/chimpanzee brain RNA-seq for correlation analysis demonstration</i>
-------------	---

---

### Description

An object of class "TekCorrSet" which contains 4 expression counts. These data are generated from speciesCounts using TEKCRABber pipeline. For a quick demo, we only select 50 orthologs and 50 transposable elements.

### Usage

```
data(speciesCorr)
```

### Format

An object of class "TekCorrSet" which contains 4 expression counts and you can access it specifying the parameters using assay\_tekcorrset():

**assay\_tekcorrset(speciesCorr, "gene", "human")** human gene expression data

**assay\_tekcorrset(speciesCorr, "te", "human")** human TE expression data

**assay\_tekcorrset(speciesCorr, "gene", "chimpanzee")** chimpanzee gene expression data

**assay\_tekcorrset(speciesCorr, "te", "chimpanzee")** chimpanzee TE expression data

### Examples

```
data(speciesCorr)
hmGeneCorrInput <- assay_tekcorrset(speciesCorr, "gene", "human")
hmTECorrInput <- assay_tekcorrset(speciesCorr, "te", "human")
chimpGeneCorrInput <- assay_tekcorrset(speciesCorr, "gene", "chimpanzee")
chimpTECorrInput <- assay_tekcorrset(speciesCorr, "te", "chimpanzee")
```

---

speciesCounts	<i>Gene/TE expression data from human/chimpanzee brain RNA-seq</i>
---------------	--

---

### Description

Dataset contains 4 expression data from human and chimpanzee brain RNA-seq. We select raw fastq data from 10 humans and 10 chimpanzees from (Khrameeva E et al., 2020). Gene expression is generated using HISAT2 and featureCounts (Kim D et al., 2019; Liao Y et al., 2014). Transposable elements (TEs) expression is generated with multi-mapping option using STAR and TEtranscripts (Dobin A et al., 2013; Jin Y et al., 2015).

**Usage**

```
data(speciesCounts)
```

**Format**

An object contains 4 expression counts:

**hmGene** human gene expression data

**hmTE** human TE expression

**chimpGene** chimpanzee gene expression data

**chimpTE** chimpanzee TE expression data

**Examples**

```
data(speciesCounts)
hmGene <- speciesCounts$hmGene
hmTE <- speciesCounts$hmTE
chimpGene <- speciesCounts$chimpGene
chimpTE <- speciesCounts$chimpTE
```

---

TEKRABber

*An R package estimates the correlations of orthologs and transposable elements between two species*

---

**Description**

TEKRABber is made to provide an user-friendly pipeline for comparing orthologs and transposable elements (TEs) between two species. It considers the orthology confidence between two species from BioMart to normalize expression counts and detect differentially expressed ortholog/TEs. Then it provides one to one correlation analysis for desired orthologs and TEs. There is also an app function to have a first insight on the result. Users can prepare orthologs/TEs RNA-seq expression data by their own preference to run TEKRABber following the data structure mentioned in the vignettes.

**Details**

TEKRABber analysis pipeline includes 5 main functions:

1. **orthologScale()**: obtain orthology information and calculate scaling factor.
2. **DECorrInputs()**: create the input files for running DE/correlation analysis.
3. **DEgeneTE()**: run DE analysis on orthologs and transposable elements.
4. **corrOrthologTE()**: estimate correlation between selected orthologs and transposable elements.
5. **appTEKRABber()**: (optional) find first insight from data using an local webapp. Find more details in vignette or on the helping page, i.e. `?orthologScale`

**Author(s)**

Yao-Chung Chen, Katja Nowick.

Maintainer: Yao-Chung Chen <yao-chung.chen@fu-berlin.de>

[TEKRABber GitHub Repo](#)

# Index

## \* datasets

- ctCorr, [5](#)
- ctInputDE, [6](#)
- fetchDataHmChimp, [9](#)
- hg38\_panTro6\_rmsk, [9](#)
- speciesCorr, [12](#)
- speciesCounts, [12](#)

- appTEKRABber, [2](#)
- assay\_tekcorrset, [4](#)

- corrOrthologTE, [4](#)
- ctCorr, [5](#)
- ctInputDE, [6](#)

- DECorrInputs, [7](#)
- DEgeneTE, [8](#)

- fetchDataHmChimp, [9](#)

- hg38\_panTro6\_rmsk, [9](#)

- orthologScale, [10](#)

- rcpp\_corr, [11](#)

- speciesCorr, [12](#)
- speciesCounts, [12](#)

- TEKRABber, [13](#)