Package ‘CodonInfo’

July 5, 2024

Type Package

Title Informational analyzes of codon frequencies for species

Version 1.0

LazyData true

Date 2024-07-05

Author Octavio Martinez

Maintainer <octavio.martinez@cinvestav.mx>

Description Using information of codon frequencies in a given species, the package estimates various informational properties of bases, codons and amino acids, providing detailed knowledge of the use of the genetic code. Estimates include Shanon's entropy (H), Effective Number of units, Information per strata under different hypotheses as well as estimation of codon bias per amino acid. This approach allows the discovery of underlying particularities in the use of the genetic code which are not obvious from the plain codon frequencies in a given species. This version contains data for 35 species, but the user can add more data for other species of interest.

License GPL-3

R topics documented:

CodonInfo-package *Informational analyzes of codon frequencies for species*

Description

Using information of codon frequencies in a given species, the package estimates various informational properties of bases, codons and amino acids, providing detailed knowledge of the use of the genetic code. Estimates include Shanon's entropy (H), Effective Number of units, Information per strata under different hypotheses as well as estimation of codon bias per amino acid. This approach allows the discovery of underlying particularities in the use of the genetic code which are not obvious from the plain codon frequencies in a given species. This version contains data for 35 species, but the user can add more data for other species of interest.

Details

The DESCRIPTION file: This package was not yet installed at build time.

Index: This package was not yet installed at build time.

Author(s)

Octavio Martinez

Maintainer: <octavio.martinez@cinvestav.mx>

References

Shannon, C.E. 1948. A mathematical theory of communication. Bell Systems Technical Journal 27:379-423; ibid 27:623-656; reprinted in C.E. Shannon: Collected Papers, N.J.A. Sloane and A.D. Wyner, eds., IEEE Press (1993).

Adami C. (2004). Information Theory in Molecular Biology. Physics of Life Reviews, Volume 1, Issue 1, April 2004, Pages 3-22.

Martinez O and Reyes-Valdes H. (2008). Defining diversity, specialization, and gene specificity in transcriptomes through information theory. PNAS, 105:28, pp. 9709–9714.

Subramanian K, Payne B, Feyertag F, and Alvarez-Ponce D (2022) The codon statistics database: a database of codon usage bias. Molecular Biology and Evolution, 39, msac157.

Martinez O. (In preparation). Sampling informational properties of codon use through the tree of life.

Examples

```
# Estimate entropy and related quantities
# in a vector of 4 relative frequencies:
est.H(c(0.2, 0.2, 0.3, 0.3))

# Estimate frequencies of bases and informational properties
# for the human frequencies of codons
round(prop.bases(nc=16, print.data=TRUE), 4)

# Estimate frequencies of bases and informational properties
# for Entamoeba histolytica
round(prop.bases(nc=10, print.data=TRUE), 4)

# Use function "prop.codons" to estimate properties at
# codon level in humans (nc=16)
temp <- prop.codons(16)
# See that this object is a list with two components:
class(temp)
# Note that the comment of this object contains the name
# of the library:
comment(temp)
# Names of the components:
names(temp)
# Summary of informational properties at codon level:
temp$H.est
# Head of the second component
head(temp$frequencies)
# Correlations between observed and expected frequencies
cor(temp$frequencies[,3:5])
# Remove the temp object
rm(temp)

# Studying codon.bias information in humans
temp <- codon.bias(nc=16)
# Gives a data.frame with results for the 19
# amino acids (aa) which can present codon bias
# (because they are coded by more than one codon)
dim(temp)
names(temp)
# Summary of main quantities
summary(temp[,4:7])
# Amino acid with smaller codon bias in humans:
temp[temp$CodBias.aa==min(temp$CodBias.aa),]
# Amino acid with larger codon bias in humans:
temp[temp$CodBias.aa==max(temp$CodBias.aa),]
# Example: aa with smaller codon bias
head(temp[order(temp$CodBias.aa), 3:7])
# Example: aa with larger codon bias
tail(temp[order(temp$CodBias.aa), 3:7])
# Remove temp object
rm(temp)
```

codon.bias — *Informational properties for amino acids coded by more than one codon*

Description

For each one of the 19 amino acids (aa) coded by more than one codon, the function estimates informational properties, including codon bias

Usage

```
codon.bias(nc = 16, only.main = FALSE)
```

Arguments

nc	Number of column of the matrix codon.freq (currently values of this parameter could be numbers 1 to 35, which determine the species to be analyzed)
only.main	logical. If only.main = FALSE only main results are obtained. Otherwise extra parameters are estimated

Details

The function calculates informational properties for all 19 cases where amino acid (aa) is coded by 2 or more codons. This includes the case of Stop (coded by codons TAA, TAG, TGA) but excludes the amino acids coded by a single codon, i.e., Met and Trp which are coded by codons ATG and TGG, respectively

Value

If only.main = FALSE (the default) a data.frame with 19 rows (amino acids, aa) and 12 columns named libray (name of the data library), fre.aa (frequency of the amino acid), aa (amino acid three letter code), n.cod (number of codons coding for the aa), H.per.aa (Entropy for the aa, calculated from the frequencies of the codons that code for that aa), H.all (Entropy for the aa, calculated from the frequencies of the bases that exist in the codons that code for that aa), Hfb (Entropy calculated using only the first base of the codons that code for that aa), Hsb (Entropy calculated using only the second base of the codons that code for that aa), Htb (Entropy calculated using only the third base of the codons that code for that aa), H.max.aa (maximum value of entropy determined by n.cod), ENC (Effective Number of Codons) and CodBias.aa (Codon Bias for the aa; a quantity that can take values from 0 -when all codons are used exactly in the same frequency, up to 1 -when only one of the codons is employed by the specie). If only.main = TRUE then only columns aa, n.cod, H.per.aa, ENC and CodBias.aa will be in the output

Author(s)

Octavio Martinez

References

Martinez O. (In preparation). Sampling informational properties of codon use through the tree of life.

See Also

codon.freq, desc.codon.freq, codons2bases, prop.bases, prop.codons

Examples

```
# Let's calculate codon bias for the case of humans:
temp <- codon.bias(nc = 16, only.main = FALSE)
dim(temp) # Number of rows and columns
# See the first row of the result
temp[1,]
# Recall which codons code for "Ala"
gen.code.n.cod[gen.code.n.cod$aa=="Ala",]
# Note that the first and second bases of those 4 codons are
# the same: "GC", thus the values of Hfb and Hsb are equal to 0.
# The entropy caused by the third base (A, C, G, T) is Htb=1.872992
# while the entropy caused by taking into account the frequencies
# of the four bases is H.all=1.635944
# By isolating the frequencies of the 4 codons in the data
# we obtained H.per.aa=1.872992 and, finally, the Codon Bias for
# Ala is obtained as (2-ENB)/(2-1), that is
(4-3.662915)/(4-1) # Approximately 0.1123617

# We could see the summary of CodBias.aa
summary(temp$CodBias.aa)
# Which is the aa with smaller Codon bias?
temp[temp$CodBias.aa == min(temp$CodBias.aa),]
# And the one with the larger Codon bias?
temp[temp$CodBias.aa == max(temp$CodBias.aa),]
# Remove the temp object
rm(temp)
```

`codon.freq` *Raw frequencies of codons in genes of 35 species*

Description

A numeric `matrix` of 64 rows (codons) and 35 columns (species) containing the raw frequencies of codons found in each one of the species. These data are the base for all calculations that could be performed in the package. In many functions to select the species that will be analyzed the only data needed is the number of column of this matrix, say, `nc`, which must be a number between 1 and 35. It is possible to add any number of data (columns) to this matrix to perform analyzes in species not already included.

Usage

```
data("codon.freq")
```

Format

Numeric matrix with the raw frequencies of codons (64 rows) in different species (35 columns). `attributes(codon.freq)$dimnames[[1]]` are the codons, `GCT`, `GCC`, etc., while `attributes(codon.freq)$dimnames` are the library names, `A.d.70779`, `A.a.5599`, etc. Those library names are formed by the first letter of the genus, the first letter of the species and the number which is the identifier of the taxa (in NCBI).

Details

Not missing data (`NA`) are present or allowed. In the cases where the Stop codons were not present (as in the case of the library `nc=11`; Enterovirus C; E.C.138950) the number of each possible stop codon was set to 1.

Source

Codon Statistics Database

References

Subramanian K, Payne B, Feyertag F, and Alvarez-Ponce D (2022) The codon statistics database: a database of codon usage bias. Molecular Biology and Evolution, 39, msac157.

Examples

```
data(codon.freq)
data(codon.freq)
# Rows and columns:
dim(codon.freq)
# Summary of sum of raw frequencies (all codons in all libraries)
summary(as.numeric(codon.freq))
# Summary of sum of raw frequencies per codon
summary(apply(codon.freq, 1, sum))
# Summary of sum of raw frequencies per library (species)
summary(apply(codon.freq, 2, sum))
# Some correlations between row frequencies
round(cor(codon.freq[,1:5]),2)
# A particularly large one
cor(codon.freq[,4], codon.freq[,5])
codon.freq[1:5, 4:5]
# Which are those species?
desc.codon.freq[4:5,]
```

---

codons2bases *Frequencies of bases in a species*

Description

Gives raw frequencies of each one of the four DNA bases (A, T, G, C) for a determined species. Optionally gives general information about the specie as well as a `data.frame` with detailed information about the frequencies of each base within each codon

Usage

```
codons2bases(nc = 16, print.data = TRUE, only.summary = TRUE)
```

Arguments

`nc`	Number of column of the matrix `codon.freq` (currently values of this parameter could be numbers 1 to 35, which determine the species to be analyzed)
`print.data`	Logic. Determines if the information about the species will be printed
`only.summary`	Logic. Determines if only the summary table will be output (when `only.summary=TRUE`) or if also a `data.frame` with details of the frequencies per codon and position within codon will be output (when `only.summary=FALSE`)

Details

The function segregates the frequencies of each DNA base by position within codons

Value

If `print.data = TRUE` and `only.summary = TRUE` then the function prints information about the data been analyzed and the output consist of a `matrix` of 4 rows and 4 columns. Columns of that matrix are labeled by the DNA bases (A, T, G, C) while the rows are labeled as F, S, T and Total, denoting the frequencies in the First, Second, Third and Total of each base in the codons of the species studied, respectively.

Else, if print.data = FALSE and only.summary = TRUE then the information about the species is not printed but the result is as described above.

Else, if print.data = TRUE and only.summary = FALSE then information about the species is printed but the result is a list of two components:

summ.tab	The matrix of frequencies described above
f.c.b	A data.frame of 64 rows and 18 columns giving frequencies for each one of the codons and columns segregating by the position of the base within each codon

Finally, if print.data = FALSE and only.summary = FALSE then no information about the specie is printed and the result is a list of two components, as described above

Note

This function is not directly useful for the user, but is employed by other functions in the package

Author(s)

Octavio Martinez

References

Martinez O. (In preparation). Sampling informational properties of codon use through the tree of life.

See Also

codon.freq, desc.codon.freq, prop.bases, prop.codons, codon.bias

Examples

```
# First, with the defaults for the function, i.e., with
# { codons2bases(nc = 16, print.data = TRUE, only.summary = TRUE) }
codons2bases()
# The function printed the information about nc=16 (data for humans)
# and the matrix in the output shows raw frequencies per base (column)
# and position within the codon. Thus for example there are
# 3086621 A's in the first position of the codons, 1985713 T's and so on.
# The last row of this matrix gives the number of bases (A, T, G, C) in
# all the three codons positions, for example there is a total of
# 8965555 A's in the genome; 3086621 in the first, 3597976 in the second
# and 2280958 in the third codon position, thus:
# 3086621 + 3597976 + 2280958 = 8965555 give the total A's in the genome.

# Now, let's try the options print.data=FALSE and only.summary=FALSE
temp <- codons2bases(nc=16, print.data=F, only.summary=F)
class(temp) # In this case we have a list with names
names(temp)
temp[[1]] # As the matrix above
# Now let's examine the second component
class(temp[[2]])
dim(temp[[2]])
names(temp[[2]])
temp[[2]][1,] # See the first row of the data.frame
# Note that adding by columns we get the same info than in temp[[1]]
# only with more details
```

```
apply(temp[[2]][, 6:18], 2, sum)
# Note about the names: fb means first base (within codon), sb means
# second base (within codon) and tb means third base (within codon)
# and compare with
temp[[1]]
# Remove the temp object
rm(temp)
```

desc.codon.freq *Description of the data in* codon.freq

Description

This data.frame depicts the main characteristics of each one of the 35 columns that contain raw frequencies for the 64 codons in the matrix of data codon.freq. This information allows the interpretation of the results of the functions in the package that analyze those raw frequencies

Usage

```
data("desc.codon.freq")
```

Format

A data frame with 35 observations on the following 12 variables.

Kingdom a character vector with the taxonomic kingdom for the species

Species a character vector with the scientific name of the species

common.name a character vector containing the common name of the species

tax.id a numeric vector with the taxonomic identifier of the species (NCBI)

key a character vector with the first two letters of each word in Species pasted with a full stop

object a character vector with the name of the object originally downloaded from the source; scientific name plus taxonomic id pasted with full stops

cod.tab a numeric vector with the identifier of the genetic code for the data

tot.nuc.gen a numeric vector with the total number of genes for the species studied

tot.cod a numeric vector with the total number of codons in the genes for the species studied

run.dat a character vector with the date (year-month-day) when the data were collected

lib.name a character vector with the library name of the data. This unique identifier is formed by the key pasted with the tax.id by a full stop. It is identical to the column names of the data matrix codon.freq

col.in.codon.freq a numeric vector with the number of column in the data matrix codon.freq

Source

Codon Statistics Database

References

Subramanian K, Payne B, Feyertag F, and Alvarez-Ponce D (2022) The codon statistics database: a database of codon usage bias. Molecular Biology and Evolution, 39, msac157.

Examples

```
data(desc.codon.freq)
# Examples of the unique identifiers for the data
head(desc.codon.freq[,c(2,4,5,11)])
# Note that desc.codon.freq$lib.name is identical to the names
# of the columns of code.freq
head(desc.codon.freq$lib.name)
# Identical to
head(attributes(codon.freq)$dimnames[[2]])

# There are 7 Kingdoms represented in this data
unique(desc.codon.freq$Kingdom)
# And here we have the table with the number of species per kingdom
table(desc.codon.freq$Kingdom)

# Some statistics for the tot.nuc.gen and tot.cod
summary(desc.codon.freq$tot.nuc.gen)
summary(desc.codon.freq$tot.cod)
cor(desc.codon.freq$tot.nuc.gen, desc.codon.freq$tot.cod)
```

est.H *Estimates informational properties from a vector of relative frequencies*

Description

From a `numeric` vector of relative frequencies gives a vector with `n` - The length of the input vector, `Hest` - The entropy (H) estimated from the frequencies, `EN` - The Effective Number of units, `Info` - The information content with reference to the maximum entropy and `V.est` - The estimated variance of the input vector.

Usage

```
est.H(x = c(3/4, 1/4), tol.in.sum = 1e-10)
```

Arguments

`x`	Must be a numeric vector of relative frequencies adding to 1. No missing values are allowed
`tol.in.sum`	Tolerance for the sum of `x`

Details

Only `x` elements larger than zero are taken into account

Value

A numeric vector with elements named as `n` <- `length(x)` (number of relative frequencies larger than zero), `Hest` <- `-sum(x*log2(x))` (estimated entropy in bits), `EN` <- `2^Hest` (Effective Number of units), `Info` <- `log2(n)-Hest` (Information with reference to the maximum entropy), `V.est` <- `var(x)` (Variance of the relative frequencies in the input `x`)

Note

This function is used to calculate informational parameters within other functions in the package

Author(s)

Octavio Martinez

References

Martinez O. (In preparation). Sampling informational properties of codon use through the tree of life.

Examples

```
# Use est.H to estimate informational parameters from the
# relative frequencies of the four bases in the human coding genome.
# First, see those frequencies (using another function)
prop.bases(nc=16)[4, 1:4]
# and now use est.H on than vector:
est.H(prop.bases(nc=16)[4, 1:4])
```

---

gen.code *The genetic code*

Description

This `data.frame` contains the (nuclear) genetic code used to classify the codons into amino acids (`aa`). Note that the three stop codons are considered to code for an amino acid

Usage

```
data("gen.code")
```

Format

A data frame with 64 observations on the following 5 variables.

`codon` a character vector with the three letter of each codon

`aa` a character vector with the three letters identifying the amino acid coded

`fb` a character vector containing the first base of the codon

`sb` a character vector containing the second base of the codon

`tb` a character vector containing the third base of the codon

Source

Genetic Code (Wikipedia)

References

Molecular Biology of the Cell. 4th edition. Alberts B, Johnson A, Lewis J, et al. New York: Garland Science; 2002.

Examples

```
data(gen.code)
# See the first rows of the data.frame
head(gen.code)
# Table the amino acids by the number of codons
table(gen.code$aa)
# Obviously every one of the four bases appears in the
# same frequency within the first, second and third positions
# within the codons:
table(gen.code$fb) # For the first base within the codon
table(gen.code$sb) # For the second base within the codon
table(gen.code$tb) # For the third base within the codon
# Codons and amino acids in which the first base is "A"
gen.code[gen.code$fb=="A",]
# Table of aa which first codon base is "A"
table(gen.code$aa[gen.code$fb=="A"])
# Codons that code for stop (here shown as an "aa")
gen.code[gen.code$aa=="Stop",]
```

`gen.code.n.cod` *Genetic code for amino acids*

Description

This `data.frame` gives the number of codons and codons coding for each one of the 21 amino acids (`aa`) in the nuclear genetic code. Note that the stop signal is taken as an amino acid

Usage

```
data("gen.code.n.cod")
```

Format

A data frame with 21 observations on the following 3 variables.

`aa` a character vector with the three code letters for the amino acid

`n.cod` a numeric vector with the number of codons coding for the corresponding amino acid (`aa`)

`codons` a character vector with the codons coding for the amino acid separated by a comma

Details

Gives the nuclear genetic code

Source

Genetic Code (Wikipedia)

References

Molecular Biology of the Cell. 4th edition. Alberts B, Johnson A, Lewis J, et al. New York: Garland Science; 2002.

Examples

```
data(gen.code.n.cod)
# See the first rows of the data.frame
head(gen.code.n.cod)
# See which amino acids are coded by a single codon
gen.code.n.cod[gen.code.n.cod$n.cod==1,]
# See that all 64 codons are included
sum(gen.code.n.cod$n.cod)
# The degeneracy of the genetic code
summary(gen.code.n.cod$n.cod)
table(gen.code.n.cod$n.cod)
# The three amino acids coded by 6 different codons
gen.code.n.cod[gen.code.n.cod$n.cod==6,]
```

prop.bases *Relative frequencies and informational properties per bases*

Description

For a given species, the function returns a data.frame with the relative frequencies of the four bases as well as informational properties. This is done for the first, second and third base of each codon as well as for the total of bases in the genome

Usage

```
prop.bases(nc = 16, print.data = FALSE, by.row = FALSE)
```

Arguments

nc	Number of column of the matrix codon.freq (currently values of this parameter could be numbers 1 to 35, which determine the species to be analyzed)
print.data	Logic. If TRUE the function prints information about the species selected to be analyzed
by.row	Logic. If TRUE the output is given as a data.frame with one row (useful to compare different species)

Value

A data.frame with 4 rows and 8 columns. The first 3 rows are named F, S, T and correspond to the First, Second and Third codon positions, while the fourth row, named Total, gives values for the total of the three bases in all codons. On the other hand, the first four columns, named A, T, G, C, denote the 4 DNA bases, while the last four columns, named Hest, ENB, Info, V.est give the informational properties obtained from the corresponding row. If by.row = TRUE the same estimates of informational properties are given formated in a single row, but frequencies of bases are excluded.

Author(s)

Octavio Martinez

References

Martinez O. (In preparation). Sampling informational properties of codon use through the tree of life.

See Also

`codon.freq`, `desc.codon.freq`, `codons2bases`, `prop.codons`, `codon.bias`

Examples

```
# Let's estimate the proportion of bases for humans
temp <- prop.bases(nc = 16, print.data = TRUE)
# See the result rounded to 4 decimals
round(temp, 4)
# Note that relative frequencies of the four bases add to 1:
apply(temp[,1:4], 1, sum)
# Also note that the last row (Total) equals the average of
# the three first rows (F, S and T)
round(apply(temp[,1:4], 2, mean), 4)
# Now columns 5 to 8 give the estimated informational properties:
# For example, taking the frequencies in the first base of codons
temp[1, 1:4]
# Gives
round(est.H(temp[1, 1:4])[2:5], 4)
# which is identical to
round(temp[1, 5:8], 4)
# Now with parameter by.row=TRUE the same output is obtained,
# except that now we have all variables in a single row:
prop.bases(nc = 16, by.row=TRUE)
# Compare with
temp[,5:8]
# Remove temp
rm(temp)
```

`prop.codons` *Estimates informational properties at codon level*

Description

For one of the species present in the `codon.freq` `data.frame`, this function calculates the observed frequencies of codons (`ObsFreq`) as well as the frequencies expected by taking into account the frequencies of the 4 DNA bases (`ExpBase`) or the frequencies of the 4 bases at each one of the three codon positions (`ExpBinC`). Using that estimates the function determine the main informational properties at codon level

Usage

```
prop.codons(nc = 16)
```

Arguments

`nc`	Number of column of the matrix `codon.freq` (currently values of this parameter could be numbers 1 to 35, which determine the species to be analyzed)

Value

A `list` with two components

`H.est` A `matrix` with 3 rows named `ObsFreq`,`ExpBase` and `ExpBinC` and 4 columns with informational properties (`Hest`, `ENC`, `Info`, `V.est`) at codon level

`frequencies` A `data.frame` with 64 rows (one for each codon) and columns `codon` (the codon), `aa` (the amino acid coded by codon), `ObsFreq` (the observed frequency of the codon in the data), `ExpBase` (the frequency of the codon expected from the frequencies of the bases in the genome) and `ExpBinC` (the frequency of the codon expected from the frequencies of the bases in each one of the three codon positions)

Author(s)

Octavio Martinez

References

Martinez O. (In preparation). Sampling informational properties of codon use through the tree of life.

See Also

`codon.freq`, `desc.codon.freq`, `codons2bases`, `prop.bases`, `codon.bias`

Examples

```
# Let's perform the analyses for humans
temp <- prop.codons(nc=16)
# Note that the comment in this object is the name of data:
comment(temp)
class(temp)
names(temp)
# Let's see the first component of the result (H.est = temp[[1]])
temp[[1]]
# And have a look at the first rows of the second (frequencies = temp[[2]])
head(temp[[2]])
# Note how the first row of temp[[1]] can be obtained:
round(est.H(temp[[2]]$ObsFreq/sum(temp[[2]]$ObsFreq)),5)
# and also the second and third rows of temp[[1]] (exercise)
# Remove temp
rm(temp)
```

Index