Package 'MetaEntropy'

May 7, 2026

Title Functional Shannon Entropy for Virome Mutational Analysis

Version 1.3

Date 2026-02-09

Description Estimates Shannon entropy, per gene and per genomic position, associated with non-synonymous mutation frequencies in viral populations, such as wastewater samples. The package uses codon translations for functional insights. Each amino acid can be treated as an individual state, resulting in a 20-state entropy computation, or grouped into one of six physicochemical classes, adding further functional context. Provides normalized values (0-1 scale) to facilitate the direct comparison of different genomic positions or total functional entropy across multiple metagenomes. Designed to analyze mutational data using tabular 'Single Nucleotide Variant' (SNV) frequency tables generated by variant callers (e.g., 'iVar' or 'LoFreq'), operating independently of consensus sequence estimation and multiple sequence alignment.

Encoding UTF-8

Depends R (>= 4.1.0)

LazyData true

Suggests rmarkdown

Imports ggplot2, patchwork, beeswarm, knitr

VignetteBuilder knitr

RoxygenNote 7.3.3

License MIT + file LICENSE

NeedsCompilation no

Author Leandro Roberto Jones [aut, cre] (ORCID: <https://orcid.org/0000-0002-5877-4194>), Julieta Marina Manrique [aut] (ORCID: <https://orcid.org/0000-0001-8712-6666>)

Maintainer Leandro Roberto Jones <lrj000@gmail.com>

Repository CRAN

Date/Publication 2026-03-09 20:20:02 UTC

Contents

`as.data.frame.entropyProfile`
Coerce `entropyProfile` *to a Data Frame*

Description

Function to extract summary information from an `entropyProfile` object. This function is internally used for plotting.

Usage

```
## S3 method for class 'entropyProfile'
as.data.frame(x, row.names = NULL, optional = FALSE, ...)
```

Arguments

`x`	An object of class `entropyProfile`.
`row.names`	Please see `as.data.frame`.
`optional`	Please see `as.data.frame`.
`...`	Additional arguments passed to the function.

Value

A `data frame` with tabular information on an entropy profile. This information includes the name of the proteins presenting mutations, the corresponding genomic positions, and the resulting entropies in the metagenome.

assessHotSpot *Evaluates Entropy Hotspot*

Description

Graphical and formal analyses of contiguous amino acids.

Usage

```
assessHotSpot(profile, boundaries, chartType = "boxplot")
```

Arguments

profile	An object of class entropyProfile.
boundaries	Numeric vector with the first and last genomic positions of the region to be evaluated. To be set interactively if not provided.
chartType	Chart type; either "boxplot", "stripchart" or "swarm".

Details

The query stretch (*e.g.* a protein domain with neutralizing epitopes) is compared against the full set of proteins. Hot spot boundaries should be indicated relative to the reference genome used in variant calling.

Value

htest object. This function is called primarily for its side effects.

See Also

getEntropySignature.

Examples

```
omicron <- getEntropySignature(wWater[wWater$wave == "third", ])

# Entrpy hotspot at SARS-CoV-2 receptor binding domain
assessHotSpot(omicron, c(22517, 23186), chartType = "swarm")
```

entropyProfile *Create (empty) object of class "entropyProfile"*

Description

This function is intended primarily for internal use by `getEntropySignature`.

Usage

```
entropyProfile(
  polymorphisms,
  position = "position",
  linkage = "linkage",
  ref = "ref",
  alt = "alt",
  protein = "protein",
  aa_position = "aa_position",
  ref_aa = "ref_aa",
  alt_aa = "alt_aa",
  alt_aa_freq = "alt_aa_freq",
  entropies = NA_real_,
  genome = mn908947.3
)
```

Arguments

`polymorphisms`	A data frame. Please see Details and Examples in documentation for `getEntropySignature`.
`position`	Name of the `polymorphisms`'s column that indicates SNV locations in the genome.
`linkage`	Information on linked positions.
`ref`	Column name with reference bases.
`alt`	Column name with the alternative bases observed in the metagenome.
`protein`	Name of the column carrying protein names.
`aa_position`	Name of the column that indicates the protein positions of the mutated amino acids.
`ref_aa`	Name of the column that carries the reference amino acids.
`alt_aa`	Name of the column carrying alternative amino acids observed in the metagenome.
`alt_aa_freq`	Name of the column giving the frequencies of alternative amino acids in the metagenome.
`entropies`	`NA_REAL_` (double numeric/real vector to hold entropy values).
`genome`	A list providing CDS data and length of the reference genome.

Details

The documentation for `getEntropySignature` details the type of input needed to create a profile. `entropyProfile` uses the same parameters as `getEntropySignature`, with the exception of `categories` and `entropies`.

Value

An (empty) object of class `entropyProfile`.

See Also

`getEntropySignature`.

`getEntropySignature` *Infer Entropy Signature*

Description

Calculates genome-wide Shannon entropies from SNV data.

Usage

```
getEntropySignature(
  polymorphisms,
  position = "position",
  linkage = "linkage",
  ref = "ref",
  alt = "alt",
  protein = "protein",
  aa_position = "aa_position",
  ref_aa = "ref_aa",
  alt_aa = "alt_aa",
  alt_aa_freq = "alt_aa_freq",
  categories = "robust",
  genome = mn908947.3
)
```

Arguments

`polymorphisms`	A data frame. Please see Details and Examples.
`position`	Name of the `polymorphisms`'s column that indicates SNV locations in the genome.
`linkage`	Information on linked positions.
`ref`	Column name with reference bases.
`alt`	Column name with the alternative bases observed in the metagenome.
`protein`	Name of the column carrying protein names.

`aa_position`	Name of the column that indicates the protein positions of the mutated amino acids.
`ref_aa`	Name of the column that carries the reference amino acids.
`alt_aa`	Name of the column carrying alternative amino acids observed in the metagenome.
`alt_aa_freq`	Name of the column giving the frequencies of alternative amino acids in the metagenome.
`categories`	Whether a class per amino acid should be used ("sensitive") or they should be grouped into aliphatic, aromatic, polar, positively charged, negatively charged, and special ("robust") (Mirny and Shakhnovich, 1999).
`genome`	A list providing CDS data and length of the reference genome.

Details

You provide a data frame with SNVs information including reference and alternative aminoacids, their frequencies, and corresponding positions relative to a reference sequence. This type of data can be generated by numerous programs and pipelines. The objective is to assess the biological impact of nonsynonymous variation within a viral population, such as an environmental sample (e.g. wastewater) or a single infection (aka quasisepecies). Entropy is calculated *within* the metagenome and is therefore independent of the reference sequence. Some mutations may be part of a same codon. This is to be indicated in the `linkage` column, providing a downstream linked position, or the closest upstream position if there are no downstream positions that are part of the same codon. For example, in the `wWater` dataset, mutations T22673C and C22674T are linked to each other and affect codon 371 of the S gene:

	wave	position	linkage	ref	alt	protein	...
...							
105	third	22599	NA	G	A	S	...
106	third	22673	22674	T	C	S	...
107	third	22674	22673	C	T	S	...
108	third	22679	NA	T	C	S	...
...							

The `genome` parameter is a list that provides data on the topology of protein-coding regions in the genome and its length, used internally primarily for graphical and summary purposes. The package provides an example (`mn908947.3`) of how this information is to be organized.

Value

An object of class `entropyProfile`. It contains a tidy, summarized version of the SNV table, a `data frame` with information on genome-wide entropy, a `data frame` with information on each CDS and corresponding mutations observed in the virome, and a `list` with CDS data and length of the reference genome used in variant calling.

References

Mirny and Shakhnovich, 1999. J Mol Biol 291:177-196. doi:10.1006/jmbi.1999.2911.

Shannon, 1948. Bell System Technical Journal, 27:379-423. doi:10.1002/j.15387305.1948.tb01338.x.

Examples

```
# Entropy across the genome in ancestral lineages
ancestral <- getEntropySignature(wWater[wWater$wave == "first", ], categories = "sensitive")

# Inspect profile
plot(ancestral, chartType = "entroScan")
```

`mn908947.3` *CDS topology and length of Wuhan-Hu-1 reference strain*

Description

This type of data can be obtained from `.gff` (General Feature Format) files using applications such as the `rtracklayer` package, or manually from the corresponding entry in the GenBank database.

Usage

```
mn908947.3
```

Format

An object of class `list` of length 2.

Source

Nucleotide [Internet]. Bethesda (MD): National Library of Medicine (US), National Center for Biotechnology Information; [1988] – . Accession No. MN908947.3, Severe acute respiratory syndrome coronavirus 2 isolate Wuhan-Hu-1, complete genome. Available from: https://www.ncbi.nlm.nih.gov/nuccore/MN908947.3.

`plot.entropyProfile` *Plot entropy signatures*

Description

Creates entropy charts along a genome.

Usage

```
## S3 method for class 'entropyProfile'
plot(x, chartType = "bp", ...)
```

Arguments

x	Object of class entropyProfile.
chartType	Whether to graph per-protein summaries ("bp"), per-protein stripcharts ("stripchart" / "swarm"), or position-wise entropy ("entroScan").
...	Additional arguments passed to the function.

Value

Unrendered gg/ggplot object produced by ggplot2. This function is primarily called for its side effects.

Examples

```
ancestral <- getEntropySignature(wWater[wWater$wave == "first", ])
omicron <- getEntropySignature(wWater[wWater$wave == "third", ])

# Enhanced Spike entropy plus pervasive negative selection in Omicron
# sublineages
anc_plot <- plot(ancestral, chartType = "stripchart")
omi_plot <- plot(omicron, chartType = "stripchart")
patchwork::wrap_plots(anc_plot/omi_plot)
```

---

print.profileSummary *Print method for* profileSummary *objects*

Description

This function formats and prints compact entropy profile summaries (profileSummary objects), on the console.

Usage

```
## S3 method for class 'profileSummary'
print(x, ...)
```

Arguments

x	An object of class profileSummary created by summary.entropyProfile.
...	Additional arguments passed to the function.

Value

Invisibly returns NULL. This function is used for its side effect.

print.tidyMutations *Print method for* tidyMutations *objects*

Description

This function formats and prints compact mutation summaries (tidyMutations objects), on the console.

Usage

```
## S3 method for class 'tidyMutations'
print(x, ...)
```

Arguments

x	An object of class tidyMutations created by showMutations.
...	Additional arguments passed to the function.

Value

Invisibly returns NULL. Called for side effect.

showMutations *Summarize mutations*

Description

Displays SNVs, and corresponding protein mutations, at specific genomic positions.

Usage

```
showMutations(profile, positions)
```

Arguments

profile	An object of class entropyProfile.
positions	A vector with genome positions relative to the reference genome.

Details

The user provides a list of genome positions and the function prints the mutations associated with them. The output format is "ref_res###alt_res / protein:ref_res###alt_res", where ref_res is the residue (eiter nucleotide or aminoacid) in the reference strain, alt_res is the alternative residue in the metagenome, "###" is the position (either nucleotide or aminoacid) where the mutation was observed, and "protein" is the name of the affected protein.

Value

An object of class `c("tidyMutations", "data.frame")`, containing summary information about user-supplied genomic positions. This information includes the mutations themselves relative to the reference genome, their positions within it, and the corresponding abundances in the virome. Intended to be displayed by `print.tidyMutations`.

See Also

`getEntropySignature`.

Examples

```
# High entropy at the RBD in Omicron lineages
omicron <- getEntropySignature(wWater[wWater$wave == "third", ])
plot(omicron, chartType="stripchart")

# Identify the high-entropy positions
omicron$Entropy$position[ omicron$Entropy$entropy > 0.3 ]
#[1] 22882 22898 22917 23013 23040 23048 23055 23063

# Get a descriptive table
showMutations(omicron, c(22882, 22898, 22917, 23013, 23040, 23048, 23055, 23063))
```

`summary.entropyProfile`

Summarize entropy profile

Description

Prints a report about an entropy profile (an object of class "entropyProfile").

Usage

```
## S3 method for class 'entropyProfile'
summary(object, ...)
```

Arguments

`object`	An object of class `entropyProfile`.
`...`	Other parameters passed to the function.

Value

An object of class `c("profileSummary", "list")` summarizing an entropy profile. Intended to be displayed via `print.profileSummary`.

wWater	*Data from first and third COVID-19 waves in Trelew* http://tools.wmflabs.org/geohack/geohack.php?language=es&pagename=Trelew&params=-43.253333333333_N_-65.309444444444_E_type:city

Description

SNVs inferred from Illumina (2 x 150) sequences from pooled ultra-pure virus concentrates representative of the 1st and 3rd COVID-19 waves in Trelew. Reads were mapped against the Wuhan-Wu-1 reference genome (MN908947.3) by *bwa*, and variants were called with *iVar* with a 3% frequency cutoff for minor variants. First wave cases were caused by ancestral strains whereas third wave cases were mainly due to highly human-adapted Omicron sublineages.

Usage

```
wWater
```

Format

An object of class `data.frame` with 148 rows and 10 columns.

Source

Manrique, Julieta Marina, and Leandro Roberto Jones. 2025. A Cost-Effective Wastewater-Based Workflow for Community-Level Insights into SARS-CoV-2 Evolution. Unpublished 0 (0): 000-000.

Index