Package: ggmRSCU (via r-universe)

May 21, 2026

Type Package

Title Visualizing Multi-Species Relative Synonymous Codon Usage and Extensible Data Exploration

Version 0.1.0

Author Xunhuan Ding [aut, cre, cph], Mingru Fu [ctb], Guojun Xing [ctb], Xinxin Ding [ctb], Xiaoli Liu [aut, ths], Tao Sun [aut, ths]

Maintainer Xunhuan Ding <xunhuanding@163.com>

Description Facilitates efficient visualization of Relative Synonymous Codon Usage patterns across species. Based on analytical outputs from 'codonW', 'MEGA', and 'Phylosuite', it supports multi-species 'RSCU' comparisons and allows users to explore visual analysis of structurally similar datasets.

License GPL (>= 3)

Encoding UTF-8

LazyData true

RoxygenNote 7.3.2

Imports data.table, dplyr, ggplot2, patchwork, purrr, rlang, stats, tidyr, utils

Depends R (>= 3.5.0)

NeedsCompilation no

Repository https://cran.r-universe.dev

Date/Publication 2025-06-24 09:10:02 UTC

RemoteUrl https://github.com/cran/ggmRSCU

RemoteRef HEAD

RemoteSha 3f71f317695edd1f4750d4d49cc3f2c1799a36ee

Contents

`energy_data` *Energy Production Projection Data*

Description

A dataset containing projected energy production values under different scenarios from 2025 to 2050. The data includes projections for various energy sources.

Usage

```
data(energy_data)
```

Format

A data frame with 24 rows and 7 variables:

Year Integer year of projection (2025-2050)

Scenario Character: High, Low, Moderate, Reference

Biomass Numeric biomass energy proportion

Coal Numeric coal energy proportion

Electrolysis Numeric electrolysis energy proportion

Gas Numeric gas energy proportion

Nuclear Numeric nuclear energy proportion

Source

Data simulated from:
Ampah, J.D., Jin, C., Liu, H. et al. (2024) "Deployment expectations of multi-gigatonne scale carbon removal could have adverse impacts on Asia's energy-water-land nexus" <doi:10.1038/s41467-024-50594-5>

financial_data *Monthly financial data (income and expenses breakdown)*

Description

Monthly financial data (income and expenses breakdown)

Usage

```
data(financial_data)
```

Format

A data frame with 66 rows and 4 variables:

Month Month of record (e.g., Jan, Feb)

Category Main category: Income or Expenses

Subcategory Detailed type (e.g., Revenue, COGS, Payroll, R&D)

Amount Amount in local currency

Source

Self-created simulated data.

genomic_data *Hominid Genomic Architecture Dataset*

Description

Region-specific genome size data for six hominid species across two haplotypes.

Usage

```
data(genomic_data)
```

Format

Tibble with 62 rows × 4 columns:

Species Species code (PAB, PPY, GGO, HSA, PPA, PTR)

group Haplotype (H1/H2)

Region Genomic region: Euchromatin, Centromere, Acrocentric, QH_region, etc.

MB Region size in megabases

Source

Data simulated from:
Yoo, D., Rhie, A., Hebbar, P. et al. (2025) "Complete sequencing of ape genomes" <doi:10.1038/s41586-025-08816-3>

`ggmRSCU` *Visualization of RSCU (Relative Synonymous Codon Usage) with ggplot2*

Description

This function generates a customized visualization of Relative Synonymous Codon Usage (RSCU) across distinct biological categories. It represents codon usage through color-coded blocks using species-specific shapes, and optionally incorporates a secondary plot displaying codon labels. Also applicable for extended visualizations of similar data structures (e.g., energy consumption data).

Usage

```
ggmRSCU(
  data,
  x,
  y,
  fill,
  rev = FALSE,
  theme = "theme1",
  shape = FALSE,
  size = 1,
  width = 0.1,
  sub_axis = "fill",
  merge = FALSE,
  border_color = NA,
  blocks_colors = NULL
)
```

Arguments

`data`	A data frame containing the input data.
`x`	Primary and secondary x-axis variables: a character vector (e.g., c("Group", "Subgroup")) or a single variable ("Group").
`y`	Primary and secondary y-axis variables: a character vector (e.g., c("ValueVar", "Subgroup")) or a single variable ("ValueVar").
`fill`	Variable for color filling.
`rev`	(Optional) logical value. If TRUE, the color order will be reversed. The default is FALSE.

theme	(Optional) A string specifying the color theme for the blocks. Available themes are: "theme1"-"theme8".
shape	(Optional) logical value. If TRUE, a vector of shape values for species in the plot is applied (default: FALSE).
size	(Optional) Size of shape markers (default: 1).
width	(Optional) Thickness of bar plot borders. (e.g., 0.1 for thin, 0.5 for thick). (Default: 0.1).
sub_axis	(Optional) Annotation type for secondary plot ("ysub" or "fill", default: "fill").
merge	(Optional) logical value. If TRUE, the main plot will be merged with the secondary annotation plot. Default is FALSE.
border_color	(Optional) Color for bar borders (default: NA)
blocks_colors	(Optional) A custom color vector to fill the plot. If "NULL", the default color scheme will be used.

Details

Key features: - Hierarchical grouping with automatic position calculation - Flexible color management with 8 predefined themes or custom palettes - Dual-axis annotation system for biological information - Interactive subplot arrangements with smart label positioning

Value

A ggplot object or patchwork composite plot when merge=TRUE

Examples

```
# Example 1: Basic Usage

library(ggmRSCU)

ggmRSCU(
  data = rscu_data,
  x = c(AA, Species),
  y = c(RSCU, Codon),
  fill = Fill,
  shape = TRUE,
  merge = TRUE
)


# Example 2: Visualize RSCU from data loaded using `read_codonw` function

# Load data from `read_codonw` output
example_dir <- system.file("extdata", "codonw", package = "ggmRSCU")
data <- read_codonw(example_dir)

col <- c("#FF6F61", "#6B5B95", "#88B04B",
         "#F7CAC9", "#92A8D1", "#9b59b6")
```

```
ggmRSCU(
  data = data,
  x = c(AA, Species),
  y = c(RSCU, Codon),
  fill = Fill,
  blocks_colors = col,
  shape = TRUE,
  merge = TRUE,
  sub_axis = "fill"
)



# Example 3: Visualize RSCU from data loaded using `read_mega` function

# Load data from `read_mega` output
example_dir <- system.file("extdata", "mega", package = "ggmRSCU")
df <- read_mega(example_dir)

ggmRSCU(
  data = df,
  x = c(AA, Species),
  y = c(RSCU, Codon),
  fill = Fill
)



# Example 4: Additional Dataset (scoring_data)

col <- c("#4E79A7", "#A0CBE8", "#F28E2B",
         "#FFBE7D", "#59A14F", "#9b59b6")

ggmRSCU(
  data = scoring_data,
  x = c(Sample, Replicate),
  y = c(Score, CellType),
  fill = CellType,
  blocks_colors = col,
  shape = TRUE,
  sub_axis = "Fill"
)


# Example 5: Additional Dataset (hydrogen_data)

library(tidyr)

df_long <- hydrogen_data %>%
  pivot_longer(cols = -c(Year, type),
  names_to = "category",
  values_to = "value")
```

```
col <- c("#4E79A7", "#A0CBE8", "#F28E2B",
         "#FFBE7D", "#59A14F", "#9b59b6")

ggmRSCU(
  data = df_long,
  x = c(Year, category),
  y = c(value, type),
  fill = type,
  shape = TRUE,
  blocks_colors = col
)


# Example 6: Additional Dataset (energy_data)

long_data <- energy_data %>%
  pivot_longer(cols = 3:7,
  names_to = "Source",
  values_to = "Production")

col <- c("#4E79A7", "#A0CBE8", "#F28E2B",
         "#FFBE7D", "#59A14F", "#9b59b6")

ggmRSCU(
  data = long_data,
  x = c(Scenario, Year),
  y = c(Production, Source),
  fill = Source,
  blocks_colors = col,
  shape = TRUE
)


# Example 7: Additional Dataset (financial_data)

ggmRSCU(
  data = financial_data,
  x = c(Month, Category),
  y = c(Amount),
  fill = Subcategory
)


# Example 8: Additional Dataset (genomic_data)

ggmRSCU(
  data = genomic_data,
  x = c(Species, haplotypes),
  y = c(Mb),
  fill = Region,
  shape = TRUE
)
```

hydrogen_data *Hydrogen production by technology and scenario (2010–2070)*

Description

Hydrogen production by technology and scenario (2010–2070)

Usage

```
data(hydrogen_data)
```

Format

A data frame with 27 rows and 8 variables:

Year Year of production

type Production technology (e.g., CG, SMR, EL, BG+CCS, etc.)

NPI Production under the NPI scenario

C Production under the C scenario

MNET Production under the MNET scenario

gCCS Production under the gCCS scenario

PBIO Production under the PBIO scenario

all Production under the all scenario

Source

Data simulated from:
Zanon-Zotin, M., Baptista, L.B., Draeger, R. et al. (2024) "Unaddressed non-energy use in the chemical industry can undermine fossil fuels phase-out" <doi:10.1038/s41467-024-52434-y>

read_codonw *Process .blk files from CodonW*

Description

This function processes .blk files from the CodonW software. It reads the files, cleans the data, and returns a combined data frame.

Usage

```
read_codonw(folder_path)
```

Arguments

folder_path	The path to the folder containing .blk files to process.

Value

A combined data frame of all processed .blk files with columns:

AA	Amino acid abbreviation
Codon	DNA codon sequence
RSCU	Relative Synonymous Codon Usage value
Fill	Position index within amino acid group
Species	Species name derived from file name

Examples

```
# Using example data
example_dir <- system.file("extdata", "codonw", package = "ggmRSCU")
result <- read_codonw(example_dir)
head(result)
```

---

read_mega *Process MEGA CSV Files*

Description

Reads and processes MEGA-generated CSV files from a specified directory, extracts codon usage data, and returns a consolidated dataset.

Usage

```
read_mega(folder_path)
```

Arguments

folder_path	Path to directory containing MEGA CSV files. Files should follow MEGA format with columns for Codon, Count, and RSCU values.

Details

Processes files through these steps:

1. Identifies valid header lines containing "Codon", "Count", and "RSCU"
2. Extracts codon/amino acid data below headers
3. Calculates position indices (Fill) for synonymous codons
4. Combines all species data into a single data frame

Value

A data frame containing:

AA Amino acid abbreviation

Codon DNA codon sequence

Fill Position index within amino acid group

Species Species identifier from file name

RSCU Calculated Relative Synonymous Codon Usage value

Examples

```
# Using example data
example_dir <- system.file("extdata", "mega", package = "ggmRSCU")
result <- read_mega(example_dir)
head(result)
```

`read_phy` *Process PhyloSuite RSCU Data for Visualization*

Description

Processes Relative Synonymous Codon Usage (RSCU) data from PhyloSuite CSV files, preparing it for visualization with the 'ggmRSCU' package.

Usage

```
read_phy(folder_path)
```

Arguments

`folder_path`	Path to folder containing CSV files. Files must contain: - 'RSCU' column - Species-reflecting file names - Required columns: AA, Codon, Fill, RSCU

Details

Performs:

- File validation for required structure
- Species name extraction
- Data reshaping to long format
- NA removal

Value

Visualization-ready data frame with:

AA Amino acid code

Codon RNA triplet

Fill Position index (1-6)

Species Organism identifier

RSCU Normalized usage values

Examples

```
# Using example data
example_dir <- system.file("extdata", "phy", package = "ggmRSCU")
result <- read_phy(example_dir)
head(result)
```

`rscu_data` *Relative Synonymous Codon Usage (RSCU) Data*

Description

A longitudinal dataset documenting codon usage bias patterns across multiple species. Contains normalized RSCU values calculated from protein- coding sequences.

Usage

```
data(rscu_data)
```

Format

A tibble with 186 rows and 5 columns:

AA `character` Standard three-letter amino acid code with isotype variants indicating synonymous codon groups: e.g., `"Phe"`, `"Leu1"`, `"Leu2"`

Codon `character` RNA codon triplet in uppercase: e.g., `"UUU"`, `"CUA"`

Fill `integer` Group index classifying synonymous codons for each amino acid (e.g., 1, 2, etc.)

Species `character` Organism identifier in `[Accession].[Version]` format: e.g., `"MK693137.1"`

RSCU `numeric` Relative Synonymous Codon Usage value

Source

Data derived from previously published studies based on protein- coding sequences analyzed in our laboratory. Accession numbers correspond to NCBI GenBank.

scoring_data *Cell Scoring Data*

Description

Simulated dataset containing normalized immune cell type scores across biological samples with technical replicates.

Usage

```
data(scoring_data)
```

Format

A data frame with 207 rows and 5 variables:

Sample Character vector indicating sample groups (e.g., "A-CP4-T", "CP4", "NTC")

Replicate Character vector identifying technical replicates (Rep1-Rep6)

Total_Score Numeric total score per replicate for normalization

CellType Character vector specifying immune cell types

Score Numeric normalized cell type score

Source

Self-created simulated data.

Index