

Package ‘iSTAY’

June 17, 2026

Type Package

Title Information-Based Stability and Synchrony Measures

Version 1.1.0

Date 2026-06-15

Author Anne Chao [aut, cre],
Yu-Ti Lee [aut],
Keng-Lei Lin [aut],
Po-Yen Chuang [aut]

Maintainer Anne Chao <chao@stat.nthu.edu.tw>

Description Provides functions to compute a continuum of information-based measures for quantifying the temporal stability of populations, communities, and ecosystems, as well as their associated synchrony, based on species (or species assemblage) biomass, or other key variables. When biodiversity data are available, the package also enables the assessment of the corresponding diversity–stability and diversity–synchrony relationships. All measures are applicable in both temporal and spatial contexts. The theoretical and methodological background is detailed in Chao et al. (2025)
<[doi:10.1101/2025.08.20.671203](https://doi.org/10.1101/2025.08.20.671203)>.

License GPL (>= 3)

Imports ggplot2 (>= 3.5.0), stringr, lmerTest, lme4, dplyr, ggpubr

Encoding UTF-8

LazyData true

Depends R (>= 4.1.0)

Suggests knitr, rmarkdown

VignetteBuilder knitr

RoxygenNote 7.3.3

NeedsCompilation no

Repository CRAN

Date/Publication 2026-06-17 17:30:02 UTC

Contents

iSTAY-package	2
Data_Jena_20_metacommunities	4
Data_Jena_462_populations	4
Data_Jena_76_community_populations	5
Data_Jena_hierarchical_structure	5
ggiSTAY_analysis	6
ggiSTAY_qprofile	9
iSTAY_Hier	12
iSTAY_Multiple	13
iSTAY_Single	15

Index	17
--------------	-----------

iSTAY-package	<i>Information-based stability and synchrony measures.</i>
---------------	--

Description

iSTAY (information-based stability and synchrony measures) is an R package that provides functions for computing a continuum of information-based measures to quantify the temporal stability of populations, communities, and ecosystems, as well as their associated synchrony, based on species (or species-assemblage) biomass or other key variables. When biodiversity data are available, the package also enables the assessment of diversity-stability and diversity-synchrony relationships. The information-based measures are derived from Hill numbers parameterized by an order $q > 0$; see Chao et al. (2025) for the theoretical and methodological background. All measures are illustrated using temporal biomass data from the Jena experiment (Roscher et al. 2004; Weisser et al. 2017; Wagg et al. 2022). See the iSTAY vignette for examples with output.

Specifically, iSTAY provides the following measures and analyses:

(1) **Single time series.** For a time series (or for each time series analyzed individually), iSTAY computes stability measures of order $q > 0$ and displays the corresponding stability profile. The stability profile illustrates how stability varies with the order q . When biodiversity data are available, iSTAY also assesses the diversity-stability relationship across individual time series.

(2) **Multiple time series.** For multiple time series (or for each set of time series within a collection), iSTAY computes four measures—gamma, alpha, and beta stability, together with synchrony. The package also displays the corresponding stability and synchrony profiles. Two weighting schemes are implemented: biomass-weighting (analogous to size-weighting in diversity analysis) and equal-weighting. When biodiversity data are available, iSTAY further assesses diversity-stability and diversity-synchrony relationships.

(3) **Hierarchical time series.** For hierarchical time series, iSTAY computes gamma, alpha, and beta stability, together with synchrony at each hierarchical level and provides the corresponding stability and synchrony profiles. Currently, only the equal-weighting scheme is implemented for

hierarchical analyses.

This package contains five main functions:

1. `iSTAY_Single`: Calculates stability measures of order $q > 0$ for a single time series of biomass or other relevant variables.
2. `iSTAY_Multiple`: Computes gamma, alpha, and beta stability, as well as synchrony, for multiple time series under either biomass-weighting (default) or equal-weighting schemes.
3. `iSTAY_Hier`: Computes gamma, alpha, and beta stability, as well as synchrony, for hierarchical time series at each hierarchical level.
4. `ggiSTAY_qprofile`: Generates stability and synchrony profile plots based on the output from `iSTAY_Single`, `iSTAY_Multiple` or `iSTAY_Hier`.
5. `ggiSTAY_analysis`: Generates diversity–stability and diversity–synchrony relationship plots based on the output from `iSTAY_Single` or `iSTAY_Multiple`.

References

- Chao, A., Colwell, R. K., Shia, J., Thorn, S., Yang, M.-Y., Mitesser, O., et al. (2025). A continuum of information-based temporal stability measures and their decomposition across hierarchical levels. *BioRxiv* <<https://doi.org/10.1101/2025.08.20.671203>>
- Roscher, C. Schumacher, J., Baade, J., Wilcke, W., Gleixner, G., Weisser, W. W. et al. (2004). The role of biodiversity for element cycling and trophic interactions: an experimental approach in a grassland community. *Basic and Applied Ecology*, 5, 107–121.
- Wagg, C., Roscher, C., Weigelt, A., Vogel, A., Ebeling, A., De Luca, E. et al. (2022). Biodiversity–stability relationships strengthen over time in a long-term grassland experiment. *Nature Communications*, 13, 7752.
- Weisser, W. W., Roscher, C., Meyer, S. T., Ebeling, A., Luo, G., Allan, E. et al. (2017). Biodiversity effects on ecosystem functioning in a 15-year grassland experiment: Patterns, mechanisms, and open questions. *Basic and Applied Ecology*, 23, 1–73.

`Data_Jena_20_metacommunities`*Biomass time series data of 20 sets of plots in the Jena experiment*

Description

All 76 plots are grouped into 20 metacommunities according to block identity and species-richness level. Each metacommunity consists of three or four plots (communities). All plots within a metacommunity have the same species richness but differ in species composition. This dataset includes the 22-year (2003 to 2024) biomass time series for constituent communities within each metacommunity.

Usage

```
data(Data_Jena_20_metacommunities)
```

Format

`Data_Jena_20_metacommunities` contains 20 lists; each list is a data frame with plot identifiers as rows and time points as columns.

`Data_Jena_462_populations`*Biomass time series data of 462 populations in the Jena experiment*

Description

The Jena Experiment consists of 76 plots arranged into 4 blocks, each containing 18–20 plots. Plots were sown along a gradient of 1, 2, 4, 8, or 16 species, resulting in a total of 462 populations. The biomass of each species in each plot was recorded annually from 2003 to 2024, except in 2004. This dataset contains the 21-year biomass time series for all 462 populations across 76 plots. This dataset is a population-by-time (462×21) data frame, in which each row represents a population and each column represents one year of biomass data.

Usage

```
data(Data_Jena_462_populations)
```

Format

`Data_Jena_462_populations` is a population-by-time data frame with 462 rows and 21 columns. Each row represents a population, and each column represents one year of biomass data.

`Data_Jena_76_community_populations`*Biomass time series data of 76 communities in the Jena experiment*

Description

The biomass data for all species sown within each plot are used to form 76 community datasets. Because the number of species sown per plot ranges from 1 to 16, the number of populations within a community also ranges from 1 to 16. Each community dataset contains the 21-year biomass time series for all constituent populations. The content of this dataset is identical to that of `Data_Jena_462_populations`, but for analytical convenience, it is reorganized into a list of 76 communities, each containing a population-by-time biomass data frame for its constituent populations.

Usage

```
data(Data_Jena_76_community_populations)
```

Format

`Data_Jena_76_community_populations` is a list of 76 data frames; each data frame has population identifiers as rows and time points as columns.

`Data_Jena_hierarchical_structure`*Four-level hierarchical structure of the Jena Experiment*

Description

The complete Jena dataset forms a four-level hierarchy: Level 1 (population or species), Level 2 (community or plot), Level 3 (block), and Level 4 (overall dataset). This dataset describes the four-level hierarchical structure. Each row corresponds to a population identifier (matching the corresponding row in `Data_Jena_462_populations`) and records the block, plot, and species to which that population belongs.

Usage

```
data(Data_Jena_hierarchical_structure)
```

Format

`Data_Jena_hierarchical_structure` is a population-by-level (462×3) data frame with population identifiers as rows and hierarchical levels ("block", "plot" and "species") as columns.

ggiSTAY_analysis	<i>ggplot2 extension for plotting diversity–stability and diversity–synchrony relationships.</i>
------------------	--

Description

ggiSTAY_analysis is a graphical function based on the output from the functions iSTAY_Single or iSTAY_Multiple. It generates plots showing the relationships between stability (and synchrony if multiple time series are included) and an additional variable, such as diversity or another relevant factor.

Usage

```
ggiSTAY_analysis(output, x_variable, by_group = NULL, model = "LMM")
```

Arguments

output	The output returned by iSTAY_Single or iSTAY_Multiple. It must include (or be combined with) a column corresponding to the variable specified in x_variable. If by_group is not NULL, it must also include a column corresponding to the variable specified in by_group.
x_variable	The name of the column representing the diversity variable (or another variable) to be displayed on the x-axis of the plot.
by_group	The name of the column representing a categorical grouping variable used to color points by group. The argument is required if model = "LMM", as the model uses it as random effect for both intercept and slope. The default is NULL.
model	Specifies the fitting model. Use model = "lm" for a linear model; or model = "LMM" for a linear mixed model with random effects for both intercept and slope. The default is model = "LMM".

Value

For an iSTAY_Single object, this function returns a figure showing the relationship between the diversity variable (or another variable) and stability.

For an iSTAY_Multiple object, this function returns a set of plots showing the relationships between the diversity variable (or another variable) and gamma, alpha, and beta stability, together with synchrony.

Examples

```
data("Data_Jena_20_metacommunities")
data("Data_Jena_76_community_populations")
data("Data_Jena_462_populations")
data("Data_Jena_hierarchical_structure")
```

```

## Single time series analysis
# Assessing diversity-stability relationships based on 76 individual plots
# See Example 2 in the iSTAY vignette for the output

communities_aggregated <- do.call(rbind, Data_Jena_20_metacommunities)

output_communities_aggregated_div <- iSTAY_Single(data = communities_aggregated,
  order.q = c(1,2), Alltime = TRUE)

output_communities_aggregated_div <- data.frame(output_communities_aggregated_div,
  log2_sowndiv = log2(as.numeric(do.call(rbind,
    strsplit(output_communities_aggregated_div[,1],
    "[.]+"))[,2])),
  block=do.call(rbind,
    strsplit(output_communities_aggregated_div[,1],
    "[.]+"))[,1])

ggiSTAY_analysis(output = output_communities_aggregated_div,
  x_variable = "log2_sowndiv",
  by_group = "block",
  model = "LMM")

# Assessing diversity-stability relationships based on 462 individual populations
# See Example 4 in the iSTAY vignette for the output.

individual_populations <- Data_Jena_462_populations

output_individual_populations_div <- iSTAY_Single(data = individual_populations,
  order.q = c(1,2), Alltime=TRUE)

output_individual_populations_div <- data.frame(
  output_individual_populations_div,
  log2_sowndiv = log2(as.numeric(do.call(rbind,
    strsplit(output_individual_populations_div[,1],
    "[.]+"))[,3])),
  block = do.call(rbind,
    strsplit(output_individual_populations_div[,1],
    "[.]+"))[,2])

ggiSTAY_analysis(output=output_individual_populations_div,
  x_variable="log2_sowndiv",
  by_group="block",
  model="LMM")

## Multiple time series analysis
# Assessing relationships between diversity and gamma, alpha, and beta stability,
# as well as synchrony, across 76 communities
# See Example 6 in the iSTAY vignette for the output.

```

```

# Equal-weighted analysis

communities <- Data_Jena_76_community_populations

output_communities_equal_div <- iSTAY_Multiple(
  data = communities,
  order.q = c(1, 2),
  equal_weights = TRUE,
  Alltime = TRUE)

output_communities_equal_div <- data.frame(
  output_communities_equal_div,
  log2_sowndiv = log2(as.numeric(do.call(rbind,
    strsplit(output_communities_equal_div[, 1],
    "[. _]+"))[, 3])),
  block = do.call(rbind,
    strsplit(output_communities_equal_div[, 1]
    , " _"))[, 2])

ggiSTAY_analysis(output = output_communities_equal_div,
  x_variable = "log2_sowndiv",
  by_group = "block",
  model = "LMM")

# Biomass-weighted analysis

output_communities_biomass_div <- iSTAY_Multiple(
  data = communities,
  order.q = c(1, 2),
  equal_weights = FALSE,
  Alltime = TRUE)

output_communities_biomass_div <- data.frame(
  output_communities_biomass_div,
  log2_sowndiv = log2(as.numeric(do.call(rbind,
    strsplit(output_communities_biomass_div[, 1],
    "[. _]+"))[, 3])),
  block = do.call(rbind,
    strsplit(output_communities_biomass_div[, 1]
    , " _"))[, 2])

ggiSTAY_analysis(output = output_communities_biomass_div,
  x_variable = "log2_sowndiv",
  by_group = "block",
  model = "LMM")

# Assessing relationships between diversity and gamma, alpha, and beta stability,
# as well as synchrony, across 20 metacommunities
# See Example 8 in the iSTAY vignette for the output.

# Equal-weighted analysis

```



```

metacommunities <- Data_Jena_20_metacommunities

output_metacommunities_equal_div <- iSTAY_Multiple(
  data = metacommunities,
  order.q = c(1, 2),
  equal_weights = TRUE,
  Alltime = TRUE)

output_metacommunities_equal_div <- data.frame(
  output_metacommunities_equal_div,
  log2_sowndiv = log2(as.numeric(do.call(rbind,
    strsplit(output_metacommunities_equal_div[, 1],
    "_"))[, 2])),
  block = do.call(rbind,
    strsplit(output_metacommunities_equal_div[, 1],
    "_"))[, 1])

ggiSTAY_analysis(output = output_metacommunities_equal_div,
  x_variable = "log2_sowndiv",
  by_group = "block",
  model = "LMM")

# Biomass-weighted analysis

output_metacommunities_biomass_div <- iSTAY_Multiple(
  data = metacommunities,
  order.q = c(1, 2),
  equal_weights = FALSE,
  Alltime = TRUE)

output_metacommunities_biomass_div <- data.frame(
  output_metacommunities_biomass_div,
  log2_sowndiv = log2(as.numeric(do.call(rbind,
    strsplit(output_metacommunities_biomass_div[, 1],
    "_"))[, 2])),
  block = do.call(rbind,
    strsplit(output_metacommunities_biomass_div[, 1],
    "_"))[, 1])

ggiSTAY_analysis(output = output_metacommunities_biomass_div,
  x_variable = "log2_sowndiv",
  by_group = "block",
  model = "LMM")

```

Description

ggiSTAY_qprofile is a graphical function based on the output from the function iSTAY_Single, iSTAY_Multiple or iSTAY_Hier. It generates stability (and synchrony, if multiple time series are included) profiles that depict how stability and synchrony vary with the order $q > 0$.

Usage

```
ggiSTAY_qprofile(output)
```

Arguments

output the output obtained from iSTAY_Single, iSTAY_Multiple or iSTAY_Hier.

Value

For an iSTAY_Single object, this function returns a figure showing the stability profile.

For an iSTAY_Multiple object, it returns a set of plots displaying the profiles for gamma, alpha, and beta stability, together with synchrony.

For an iSTAY_Hier object, it returns a set of plots displaying the profiles for gamma, alpha, and beta stability, together with synchrony.

Examples

```
data("Data_Jena_20_metacommunities")
data("Data_Jena_76_community_populations")
data("Data_Jena_462_populations")
data("Data_Jena_hierarchical_structure")

## Single time series analysis
# Comparing stability profiles for two selected individual plots
# See Example 1 in the iSTAY vignette for the output

communities_aggregated <- do.call(rbind, Data_Jena_20_metacommunities)

output_two_plots_q <- iSTAY_Single(data = communities_aggregated[
  which(rownames(communities_aggregated) %in% c("B1_4.B1A04", "B4_2.B4A14")),],
  order.q=seq(0.1,2,0.1),
  Alltime = TRUE)

ggiSTAY_qprofile(output = output_two_plots_q)

# Comparing stability profiles for two selected individual populations
# See Example 3 in the iSTAY vignette for the output

individual_populations <- Data_Jena_462_populations
```

```

output_two_populations_q <- iSTAY_Single(data = individual_populations[
  which(rownames(individual_populations) %in%
    c("B1A06_B1_16_BM_Ant.odo", "B1A06_B1_16_BM_Cam.pat")),],
  order.q=seq(0.1,2,0.1), Alltime = TRUE)

ggiSTAY_qprofile(output = output_two_populations_q)

## Multiple time series analysis
# Comparing gamma, alpha, and beta stability profiles,
# and synchrony profiles in two selected communities
# See Example 5 in the iSTAY vignette for the output

communities <- Data_Jena_76_community_populations

# Equal-weighted analysis

output_two_communities_equal_q <- iSTAY_Multiple(
  data = communities[which(names(communities) %in% c("B1A04_B1_4", "B4A14_B4_2"))],
  order.q = seq(0.1, 2, 0.1), equal_weights = TRUE, Alltime = TRUE)

ggiSTAY_qprofile(output = output_two_communities_equal_q)

# Biomass-weighted analysis

output_two_communities_biomass_q <- iSTAY_Multiple(
  data = communities[which(names(communities) %in% c("B1A04_B1_4", "B4A14_B4_2"))],
  order.q = seq(0.1, 2, 0.1), equal_weights = FALSE, Alltime = TRUE)

ggiSTAY_qprofile(output = output_two_communities_biomass_q)

# Comparing gamma, alpha, and beta stability profiles,
# and synchrony profiles for two selected metacommunities
# See Example 7 in the iSTAY vignette for the output.

metacommunities <- Data_Jena_20_metacommunities

# Equal-weighted analysis

output_two_metacommunities_equal_q <- iSTAY_Multiple(
  data = metacommunities[which(names(metacommunities) %in% c("B1_1", "B3_2"))],
  order.q = seq(0.1, 2, 0.1),
  equal_weights = TRUE,
  Alltime = TRUE)

ggiSTAY_qprofile(output = output_two_metacommunities_equal_q)

# Biomass-weighted analysis

```

```

output_two_metacommunities_biomass_q <- iSTAY_Multiple(
  data = metacommunities[which(names(metacommunities) %in% c("B1_1", "B3_2"))],
  order.q = seq(0.1, 2, 0.1),
  equal_weights = FALSE,
  Alltime = TRUE)

ggiSTAY_qprofile(output = output_two_metacommunities_biomass_q)

## Hierarchical time series analysis
# Plotting stability and synchrony profiles at each hierarchical level
# See Example 9 in the iSTAY vignette for the output

data("Data_Jena_462_populations")
data("Data_Jena_hierarchical_structure")

output_hier_q <- iSTAY_Hier(data = Data_Jena_462_populations,
  structure = Data_Jena_hierarchical_structure,
  order.q = seq(0.1,2,0.1), Alltime = TRUE)

ggiSTAY_qprofile(output = output_hier_q)

```

iSTAY_Hier

Calculate stability and synchrony at each hierarchical level

Description

iSTAY_Hier computes gamma, alpha, and beta stability, as well as synchrony, at each hierarchical level for time series of biomass or other variables. Currently, only the equal-weighting scheme is implemented for hierarchical analyses.

Usage

```

iSTAY_Hier(
  data,
  structure,
  order.q = c(1, 2),
  Alltime = TRUE,
  start_T = NULL,
  end_T = NULL
)

```

Arguments

data	A data.frame containing the hierarchical data, with sampling units as rows and time points as columns.
------	--

structure	The hierarchical structure of the input data.
order.q	A numerical vector specifying the orders of stability. Default is c(1,2).
Alltime	Logical (TRUE or FALSE), indicating whether to use all time points in the data.
start_T	(Applicable only if Alltime = FALSE) a positive integer specifying the starting column (time point) for the analysis interval.
end_T	(Applicable only if Alltime = FALSE) a positive integer specifying the ending column (time point) for the analysis interval.

Value

a data frame with the following columns:

Hier_level: hierarchical level (e.g. Level 1: population, Level 2: community, Level 3: block, and Level 4: overall data)

Order_q: order of stability or synchrony

Gamma, Alpha, Beta: stability measures of order q

Synchrony: synchrony measure of order q

Examples

```
data("Data_Jena_462_populations")
data("Data_Jena_hierarchical_structure")
output_hier <- iSTAY_Hier(data = Data_Jena_462_populations,
  structure = Data_Jena_hierarchical_structure,
  order.q=c(1,2), Alltime=TRUE)
output_hier
```

iSTAY_Multiple	<i>Calculate stability and synchrony for multiple time series.</i>
----------------	--

Description

iSTAY_Multiple computes gamma, alpha, and beta stability, as well as synchrony, for multiple time series data. Two weighting schemes are implemented: biomass-weighting (analogous to size-weighting in diversity analysis) and equal-weighting.

Usage

```
iSTAY_Multiple(
  data,
  order.q = c(1, 2),
  equal_weights = FALSE,
  Alltime = TRUE,
  start_T = NULL,
  end_T = NULL
)
```



```

                                Alltime = TRUE)

output_communities_biomass

# Computing gamma, alpha, and beta stability together with synchrony
# for 20 individual metacommunities under equal-weighting

data("Data_Jena_20_metacommunities")

metacommunities <- Data_Jena_20_metacommunities

output_metacommunities_equal <- iSTAY_Multiple(data = metacommunities,
                                              order.q = c(1, 2),
                                              equal_weights = TRUE,
                                              Alltime = TRUE)

output_metacommunities_equal

# Computing gamma, alpha, and beta stability together with synchrony
# for 20 individual metacommunities under biomass/size weighting

output_metacommunities_biomass <- iSTAY_Multiple(data = metacommunities,
                                              order.q = c(1, 2),
                                              equal_weights = FALSE,
                                              Alltime = TRUE)

output_metacommunities_biomass

```

iSTAY_Single	<i>Calculate stability for a single time series.</i>
--------------	--

Description

iSTAY_Single computes the stability of order q for a single time series.

Usage

```

iSTAY_Single(
  data,
  order.q = c(1, 2),
  Alltime = TRUE,
  start_T = NULL,
  end_T = NULL
)

```

Arguments

<code>data</code>	A vector of time series data, or a <code>data.frame</code> with sampling units as rows and time points as columns.
<code>order.q</code>	A numerical vector specifying the orders of stability. Default is <code>c(1,2)</code> .
<code>Alltime</code>	Logical (TRUE or FALSE), indicating whether to use all time points in the data.
<code>start_T</code>	(Applicable only if <code>Alltime = FALSE</code>) a positive integer specifying the starting column (time point) for the analysis interval.
<code>end_T</code>	(Applicable only if <code>Alltime = FALSE</code>) a positive integer specifying the ending column (time point) for the analysis interval.

Value

a data frame with the following columns:
 Dataset: the input dataset
 Order_q: order of stability
 Stability: stability measures of order q

Examples

```
# Computing stability of individual communities

data("Data_Jena_20_metacommunities")

communities_aggregated <- do.call(rbind, Data_Jena_20_metacommunities)

output_communities_aggregated <- iSTAY_Single(data = communities_aggregated,
                                              order.q=c(1,2), Alltime = TRUE)

output_communities_aggregated

# Computing stability of individual populations

data("Data_Jena_462_populations")

individual_populations <- Data_Jena_462_populations

output_individual_populations <- iSTAY_Single(data = individual_populations,
                                              order.q = c(1,2), Alltime = TRUE)

output_individual_populations
```


Index

* **datasets**

- Data_Jena_20_metacommunities, [4](#)
- Data_Jena_462_populations, [4](#)
- Data_Jena_76_community_populations,
[5](#)
- Data_Jena_hierarchical_structure,
[5](#)

* **package**

- iSTAY-package, [2](#)

- Data_Jena_20_metacommunities, [4](#)
- Data_Jena_462_populations, [4](#)
- Data_Jena_76_community_populations, [5](#)
- Data_Jena_hierarchical_structure, [5](#)

- ggiSTAY_analysis, [6](#)
- ggiSTAY_qprofile, [9](#)

- iSTAY (iSTAY-package), [2](#)
- iSTAY-package, [2](#)
- iSTAY_Hier, [12](#)
- iSTAY_Multiple, [13](#)
- iSTAY_Single, [15](#)