Package 'gp3sequences'

July 30, 2026

Title Transparent Analysis of Ordered Categorical Sequences

Version 0.1.0

Description Provides transparent, reproducible, and auditable tools for validating, preparing, encoding, and summarising ordered categorical sequence data. Supports configurable long-format inputs, explicit preprocessing policies, machine-readable diagnostics, structural motif analysis, and focused visualisation for scanpaths, navigation paths, behavioural states, and other ordered state data.

License MIT + file LICENSE

URL https://stefanosbalaskas.github.io/gp3sequences/, https://github.com/stefanosbalaskas/gp3sequences

BugReports https://github.com/stefanosbalaskas/gp3sequences/issues

Suggests testthat (>= 3.0.0), knitr, rmarkdown

Config/testthat/edition 3

Config/Needs/website pkgdown

Encoding UTF-8

RoxygenNote 8.0.0

Imports graphics, stats

VignetteBuilder knitr

NeedsCompilation no

Author Stefanos Balaskas [aut, cre] (ORCID: <https://orcid.org/0000-0003-2444-9796>)

Maintainer Stefanos Balaskas <s.balaskas@ac.upatras.gr>

Repository CRAN

Date/Publication 2026-07-30 17:20:10 UTC

Contents

`audit_sequence_data` *Audit Long-Format Sequence Data*

Description

Examines a long-format data frame against the neutral `gp3sequences` sequence-data contract without modifying the input.

Usage

```
audit_sequence_data(
  data,
  sequence_id_col,
  order_col,
  state_col,
  duration_col = NULL,
  metadata_cols = NULL,
  expected_states = NULL
)
```

Arguments

`data`	A data frame containing ordered state observations.
`sequence_id_col`	Name of the sequence identifier column.
`order_col`	Name of the numeric sequence-order column.
`state_col`	Name of the categorical state column.

`duration_col` Optional name of a numeric duration column.

`metadata_cols` Optional character vector naming columns that should remain constant within each sequence.

`expected_states` Optional vector of known or permitted state values.

Details

The audit checks column mappings, empty inputs, missing identifiers, missing or non-numeric order values, duplicated positions, integer order gaps, unordered rows, missing states, consecutive repeated states, single-row sequences, invalid durations, inconsistent metadata, unexpected states, and unused factor levels.

The function reports structural properties only. It does not infer psychological, cognitive, emotional, or diagnostic states.

Value

A data frame with one row per detected issue and the stable columns `sequence_id`, `row`, `column`, `issue_code`, `severity`, `value`, `message`, and `action`. Severity values are `error`, `review`, and `info`.

Examples

```
sequences <- data.frame(
  id = rep(c("s1", "s2"), each = 3L),
  position = rep(1:3, times = 2L),
  state = c("home", "search", "product", "home", "category", "product")
)

audit_sequence_data(
  sequences,
  sequence_id_col = "id",
  order_col = "position",
  state_col = "state"
)
```

---

encode_sequence_data *Encode Ordered Sequence States*

Description

Creates a deterministic dictionary and adds integer and labelled state codes to long-format sequence data.

Usage

```
encode_sequence_data(
  data,
  sequence_id_col,
  order_col,
  state_col,
  duration_col = NULL,
  metadata_cols = NULL,
  expected_states = NULL,
  state_levels = NULL,
  prefix = "S",
  width = NULL
)
```

Arguments

`data`	A data frame containing ordered state observations.
`sequence_id_col`	Name of the sequence identifier column.
`order_col`	Name of the numeric sequence-order column.
`state_col`	Name of the categorical state column.
`duration_col`	Optional name of a numeric duration column.
`metadata_cols`	Optional character vector naming columns that should remain constant within each sequence.
`expected_states`	Optional vector of known or permitted state values.
`state_levels`	Optional atomic vector defining the complete state ordering. When omitted, factor levels are respected; otherwise observed state labels are sorted alphabetically.
`prefix`	Character prefix used for labelled codes.
`width`	Optional positive integer width for the numeric part of each labelled code. The default is determined from the dictionary size.

Details

The function does not reinterpret states. State codes are transparent identifiers derived from an explicit or deterministic state ordering.

Value

A named list containing:

- `data`: deterministically sorted canonical data with `state_index` and `state_code`;
- `dictionary`: state labels, integer indices, labelled codes, and an observed-state indicator;
- `audit`, `status`, and `mapping` from input validation;
- `settings`: the resolved code prefix and width.

Examples

```
sequences <- data.frame(
  id = c("s1", "s1", "s2", "s2"),
  position = c(1, 2, 1, 2),
  state = c("home", "search", "home", "product")
)

encoded <- encode_sequence_data(
  sequences,
  sequence_id_col = "id",
  order_col = "position",
  state_col = "state"
)

encoded$dictionary
encoded$data
```

`extract_sequence_ngrams` *Extract Contiguous Sequence N-Grams*

Description

Enumerates contiguous state motifs from validated long-format sequence data.

Usage

```
extract_sequence_ngrams(
  data,
  sequence_id_col,
  order_col,
  state_col,
  duration_col = NULL,
  metadata_cols = NULL,
  expected_states = NULL,
  min_length = 2L,
  max_length = 3L,
  overlap = c("allow", "disallow"),
  separator = " > ",
  state_levels = NULL
)
```

Arguments

`data`	A data frame containing ordered state observations.
`sequence_id_col`	Name of the sequence identifier column.

order_col	Name of the numeric sequence-order column.
state_col	Name of the categorical state column.
duration_col	Optional name of a numeric duration column.
metadata_cols	Optional character vector naming columns that should remain constant within each sequence.
expected_states	Optional vector of known or permitted state values.
min_length	Positive whole number giving the shortest motif length.
max_length	Positive whole number giving the longest motif length.
overlap	Character value specifying whether overlapping occurrences of the same motif within the same sequence are "allow"ed or "disallow"ed.
separator	Character value used only to display state labels in the human-readable motif column.
state_levels	Optional atomic vector defining the complete state ordering. When omitted, factor levels are respected; otherwise observed labels are sorted deterministically.

Details

Motifs are contiguous windows only. No subsequence gaps, edit distances, statistical tests, or substantive interpretations are introduced. Consecutive repeated states are used exactly as supplied; any repeat collapsing should be performed explicitly with prepare_sequence_data() before extraction.

With overlap = "disallow", overlap is resolved independently within each sequence-motif pair by a deterministic left-to-right greedy rule. Different motifs and different motif lengths do not compete for positions.

The collision-resistant motif_id and motif_key columns are derived from deterministic state codes. The display separator may therefore also occur inside a state label without changing motif identity.

Value

A named list containing:

- occurrences: one row per retained contiguous motif occurrence;
- motifs: the distinct motif dictionary;
- sequences: sequence-level counts of candidate and retained occurrences;
- state_dictionary: the deterministic state dictionary;
- audit, status, and mapping from input validation;
- settings: the resolved motif extraction settings.

Examples

```
sequences <- data.frame(
  id = c(rep("s1", 5L), rep("s2", 4L)),
  position = c(1:5, 1:4),
  state = c("A", "B", "A", "B", "A", "A", "B", "A", "C")
)

ngrams <- extract_sequence_ngrams(
  sequences,
  sequence_id_col = "id",
  order_col = "position",
  state_col = "state",
  min_length = 2,
  max_length = 3,
  overlap = "allow"
)

ngrams$occurrences
ngrams$motifs
```

---

filter_sequence_motifs

Filter Sequence Motif Summaries

Description

Applies transparent count, prevalence, length, and top-ranking filters to sequence motif summaries.

Usage

```
filter_sequence_motifs(
  x,
  min_occurrences = 1L,
  min_sequences = 1L,
  min_prevalence = 0,
  motif_lengths = NULL,
  top_n = NULL,
  rank_by = c("sequence_prevalence", "n_occurrences", "n_sequences"),
  ties = c("include", "first")
)
```

Arguments

x	A motif extraction, motif summary, or filtered-motif object.
min_occurrences	Non-negative whole number giving the minimum total occurrence count.

min_sequences	Non-negative whole number giving the minimum number of sequences containing the motif.
min_prevalence	Minimum sequence prevalence between 0 and 1.
motif_lengths	Optional vector of positive whole-number motif lengths to retain.
top_n	Optional positive whole number giving the requested number of highest-ranked motifs.
rank_by	Metric used for deterministic top-ranking: one of "sequence_prevalence", "n_occurrences", or "n_sequences".
ties	Character value controlling the top-n boundary. "include" retains every motif tied on rank_by with the final selected motif; "first" retains exactly top_n motifs after deterministic secondary sorting.

Details

Filtering is descriptive and deterministic. When ties = "first", ties are resolved by sequence prevalence, occurrence count, sequence count, shorter motif length, and finally the collision-resistant motif key.

Value

A named list containing the filtered motifs table, the matching sequence-level rows in by_sequence, sequence and state dictionaries, validation metadata, filter settings, and counts before and after filtering.

Examples

```
sequences <- data.frame(
  id = c(rep("s1", 5L), rep("s2", 4L)),
  position = c(1:5, 1:4),
  state = c("A", "B", "A", "B", "A", "A", "B", "A", "C")
)

extracted <- extract_sequence_ngrams(
  sequences,
  sequence_id_col = "id",
  order_col = "position",
  state_col = "state"
)

filtered <- filter_sequence_motifs(
  extracted,
  min_sequences = 2,
  top_n = 5,
  ties = "include"
)

filtered$motifs
```

format_sequence_motifs *Format Sequence Motif Summaries*

Description

Produces a stable, report-ready table from motif extraction, summary, or filtered-motif output.

Usage

```
format_sequence_motifs(
  x,
  digits = 3L,
  prevalence = c("proportion", "percent"),
  include_rank = TRUE,
  rank_by = c("sequence_prevalence", "n_occurrences", "n_sequences"),
  ties = c("min", "first"),
  include_ids = TRUE
)
```

Arguments

x	A motif extraction, motif summary, or filtered-motif object.
digits	Whole number from 0 to 15 controlling numeric rounding.
prevalence	Character value specifying whether prevalence and occurrence share are shown as "proportion"s or "percent"ages.
include_rank	Logical value indicating whether to add a rank column.
rank_by	Metric used to order and rank motifs: one of "sequence_prevalence", "n_occurrences", or "n_sequences".
ties	Character value specifying "min" shared ranks or deterministic "first" ranks.
include_ids	Logical value indicating whether motif_id and motif_key should be included in the formatted table.

Details

Formatting changes display precision and units only. It does not change the underlying motif counts or introduce substantive interpretation.

Value

A named list containing table, validation metadata, and formatting settings. The table contains only structural motif measurements.

Examples

```
sequences <- data.frame(
  id = c(rep("s1", 5L), rep("s2", 4L)),
  position = c(1:5, 1:4),
  state = c("A", "B", "A", "B", "A", "A", "B", "A", "C")
)

extracted <- extract_sequence_ngrams(
  sequences,
  sequence_id_col = "id",
  order_col = "position",
  state_col = "state"
)

formatted <- format_sequence_motifs(
  extracted,
  prevalence = "percent",
  digits = 1
)

formatted$table
```

---

format_sequence_motif_positions

Format Sequence Motif Position Summaries

Description

Produces a deterministic report-ready table from positional motif summaries.

Usage

```
format_sequence_motif_positions(
  x,
  digits = 3L,
  position_units = c("proportion", "percent"),
  include_rank = TRUE
)
```

Arguments

x	An object returned by summarise_sequence_motif_positions().
digits	Whole number from 0 to 15 controlling numeric rounding.
position_units	Display units for relative positions: "proportion" or "percent". Absolute positions remain one-based sequence indices.
include_rank	Logical value indicating whether to include an earlier-to-later rank based on mean position. Ranking is performed within each supplied by group.

Details

Formatting copies and transforms the summary table only. The input object is not modified. Rows are ordered deterministically by grouping values, mean and median position, occurrence and sequence counts, motif length, and motif key. Equal mean positions receive the same minimum rank.

Value

A named list containing the formatted `table`, validation metadata, and formatting settings.

Examples

```
sequences <- data.frame(
  id = c(rep("s1", 5L), rep("s2", 4L)),
  position = c(1:5, 1:4),
  state = c("A", "B", "A", "B", "C", "A", "B", "C", "B")
)

extracted <- extract_sequence_ngrams(
  sequences,
  sequence_id_col = "id",
  order_col = "position",
  state_col = "state"
)

positions <- summarise_sequence_motif_positions(
  extracted,
  position = "centre",
  scale = "relative"
)

formatted <- format_sequence_motif_positions(
  positions,
  position_units = "percent",
  digits = 1
)

formatted$table
```

---

format_sequence_paths *Format Ordered Sequence Paths*

Description

Creates a compact one-row-per-sequence representation of ordered state paths.

Usage

```
format_sequence_paths(
  data,
  sequence_id_col,
  order_col,
  state_col,
  metadata_cols = NULL,
  expected_states = NULL,
  separator = " > ",
  collapse_repeats = FALSE
)
```

Arguments

`data`	A data frame containing ordered state observations.
`sequence_id_col`	Name of the sequence identifier column.
`order_col`	Name of the numeric sequence-order column.
`state_col`	Name of the categorical state column.
`metadata_cols`	Optional character vector naming columns that should remain constant within each sequence.
`expected_states`	Optional vector of known or permitted state values.
`separator`	Character value inserted between adjacent state labels.
`collapse_repeats`	Logical value indicating whether consecutive repeated states should be collapsed for path display.

Details

Repeat collapsing affects only the formatted representation. It does not modify the supplied data or alter non-consecutive repeated states. Input rows are ordered deterministically for formatting, but review-level diagnostics such as `unordered_rows` remain reflected in the returned `status` and `audit`.

Value

A named list containing:

- `paths`: one row per sequence with observation counts, formatted-state counts, unique-state counts, start and end states, and the path string;
- `audit`, `status`, and `mapping` from input validation;
- `settings`: the path separator and repeat-collapsing choice.

Examples

```
sequences <- data.frame(
  id = c("s1", "s1", "s1", "s2", "s2"),
  position = c(1, 2, 3, 1, 2),
  state = c("A", "B", "C", "A", "C")
)

paths <- format_sequence_paths(
  sequences,
  sequence_id_col = "id",
  order_col = "position",
  state_col = "state"
)

paths$paths
```

---

plot_sequence_motifs *Plot Sequence Motif Summaries*

Description

Draws a dependency-free base-R bar chart of structural motif measurements.

Usage

```
plot_sequence_motifs(
  x,
 metric = c("sequence_prevalence", "n_occurrences", "n_sequences", "occurrence_share"),
  top_n = 20L,
  motif_lengths = NULL,
  ties = c("include", "first"),
  horizontal = TRUE
)
```

Arguments

x	A motif extraction, summary, or filtered-motif object.
metric	Structural metric to plot: "sequence_prevalence", "n_occurrences", "n_sequences", or "occurrence_share".
top_n	Positive whole number giving the requested number of motifs.
motif_lengths	Optional vector of positive whole-number motif lengths to retain.
ties	Top-n boundary policy. "include" retains all motifs tied on the selected metric; "first" retains exactly top_n after deterministic secondary sorting.
horizontal	Logical value indicating whether bars should be horizontal.

Details

The plot is descriptive. It does not perform inferential testing or assign substantive meaning to motif frequency or prevalence. Empty inputs produce an informative blank plot and return an empty table.

Value

Invisibly returns the exact motif table used for plotting, including `plot_rank`, `plot_value`, `plot_label`, and `bar_midpoint`.

Examples

```
sequences <- data.frame(
  id = c(rep("s1", 5L), rep("s2", 4L)),
  position = c(1:5, 1:4),
  state = c("A", "B", "A", "B", "C", "A", "B", "C", "B")
)

extracted <- extract_sequence_ngrams(
  sequences,
  sequence_id_col = "id",
  order_col = "position",
  state_col = "state"
)

plot_sequence_motifs(
  extracted,
  metric = "sequence_prevalence",
  top_n = 10
)
```

---

plot_sequence_motif_positions

Plot Sequence Motif Positions

Description

Shows occurrence locations for selected contiguous motifs.

Usage

```
plot_sequence_motif_positions(
  x,
  motifs = NULL,
  position = c("start", "centre", "end"),
  scale = c("absolute", "relative"),
  top_n = 10L,
  display = c("strip", "distribution")
)
```

Arguments

x	An object returned by `extract_sequence_ngrams()` or `summarise_sequence_motif_positions()`.
motifs	Optional character vector of motif identifiers, motif keys, or displayed motif labels. When omitted, motifs are selected deterministically by occurrence count, sequence count, motif length, and motif key.
position	Occurrence position represented by motif `"start"`, `"centre"`, or `"end"`.
scale	Position scale: one-based `"absolute"` sequence positions or `"relative"` positions from 0 to 1.
top_n	Positive whole number giving the number of motifs selected when `motifs` is `NULL`.
display	Base-R display type: occurrence `"strip"` or `"distribution"` boxplots.

Details

The strip display uses deterministic vertical stacking rather than random jitter. The distribution display uses one horizontal boxplot per motif. Empty inputs produce an informative blank plot and return an empty table. The function provides structural location summaries only.

Value

Invisibly returns the exact occurrence-level data used by the plot, including deterministic motif ranks and plotting coordinates where relevant.

Examples

```
sequences <- data.frame(
  id = c(rep("s1", 5L), rep("s2", 4L)),
  position = c(1:5, 1:4),
  state = c("A", "B", "A", "B", "C", "A", "B", "C", "B")
)

extracted <- extract_sequence_ngrams(
  sequences,
  sequence_id_col = "id",
  order_col = "position",
  state_col = "state"
)

plot_sequence_motif_positions(
  extracted,
  position = "centre",
  scale = "relative",
  top_n = 5,
  display = "strip"
)
```

`prepare_sequence_data` *Prepare Long-Format Sequence Data*

Description

Applies explicit preprocessing policies and returns a deterministic, canonical long-format representation.

Usage

```
prepare_sequence_data(
  data,
  sequence_id_col,
  order_col,
  state_col,
  duration_col = NULL,
  metadata_cols = NULL,
  expected_states = NULL,
  missing_state_policy = c("error", "drop"),
  duplicate_position_policy = c("error", "first", "last"),
  repeated_state_policy = c("preserve", "collapse"),
  zero_duration_policy = c("preserve", "drop", "error"),
  unknown_state_policy = c("preserve", "drop", "error"),
  unused_state_levels = c("preserve", "drop")
)
```

Arguments

`data`	A data frame containing ordered state observations.
`sequence_id_col`	Name of the sequence identifier column.
`order_col`	Name of the numeric sequence-order column.
`state_col`	Name of the categorical state column.
`duration_col`	Optional name of a numeric duration column.
`metadata_cols`	Optional character vector naming columns that should remain constant within each sequence.
`expected_states`	Optional vector of known or permitted state values.
`missing_state_policy`	Policy for missing states: `"error"` or `"drop"`.
`duplicate_position_policy`	Policy for duplicated sequence positions: `"error"`, `"first"`, or `"last"`.
`repeated_state_policy`	Policy for consecutive repeated states: `"preserve"` or `"collapse"`.

`zero_duration_policy`
: Policy for zero durations: `"preserve"`, `"drop"`, or `"error"`.

`unknown_state_policy`
: Policy for states absent from `expected_states`: `"preserve"`, `"drop"`, or `"error"`.

`unused_state_levels`
: Policy for unused factor levels: `"preserve"` or `"drop"`.

Details

Rows are sorted deterministically by sequence identifier, sequence order, and original row number. When consecutive repeats are collapsed, the first row supplies non-duration values and available durations are summed.

Unresolved errors produce `status = "fail"` and `data = NULL`; diagnostics and decision records remain available.

Value

A named list containing:

- `data`: canonical prepared data, or `NULL` when unresolved errors remain;
- `audit`: input- and output-stage diagnostics;
- `decisions`: a machine-readable preprocessing decision log;
- `mapping`: source-to-contract column mappings;
- `status`: `pass`, `review`, or `fail`;
- row counts and final state levels.

The canonical columns are `sequence_id`, `sequence_order`, `state`, `original_row`, and optional `duration`. Unmapped columns are preserved.

Examples

```
sequences <- data.frame(
  id = c("s2", "s1", "s1", "s2"),
  position = c(2, 2, 1, 1),
  state = c("product", "search", "home", "home")
)

prepared <- prepare_sequence_data(
  sequences,
  sequence_id_col = "id",
  order_col = "position",
  state_col = "state"
)

prepared$status
prepared$data
```

summarise_sequence_motifs
Summarise Contiguous Sequence Motifs

Description

Aggregates extracted contiguous motif occurrences by sequence and overall.

Usage

```
summarise_sequence_motifs(x)
```

Arguments

x	An object returned by `extract_sequence_ngrams()`.

Details

Sequence prevalence uses every validated sequence in the extraction object as its denominator, including sequences too short to contain a requested motif. Results are sorted deterministically by sequence prevalence, occurrence count, motif length, and motif key.

The function reports structural recurrence only. It does not perform significance testing or infer psychological, cognitive, emotional, or diagnostic attributes.

Value

A named list containing:

- `by_sequence`: occurrence counts for each sequence-motif pair;
- `overall`: total occurrence counts, sequence counts, sequence prevalence, occurrence share, and mean occurrence rates;
- `sequences` and `state_dictionary` from extraction;
- `audit`, `status`, `mapping`, and extraction settings;
- scalar counts for sequences, occurrences, and distinct motifs.

Examples

```
sequences <- data.frame(
  id = c(rep("s1", 5L), rep("s2", 4L)),
  position = c(1:5, 1:4),
  state = c("A", "B", "A", "B", "A", "A", "B", "A", "C")
)

extracted <- extract_sequence_ngrams(
  sequences,
  sequence_id_col = "id",
  order_col = "position",
```

```
  state_col = "state",
  min_length = 2,
  max_length = 3
)

summaries <- summarise_sequence_motifs(extracted)
summaries$by_sequence
summaries$overall
```

`summarise_sequence_motif_positions` *Summarise Sequence Motif Positions*

Description

Summarises where contiguous motif occurrences appear within sequences.

Usage

```
summarise_sequence_motif_positions(
  x,
  position = c("start", "centre", "end"),
  scale = c("absolute", "relative"),
  by = NULL
)
```

Arguments

`x`	An object returned by `extract_sequence_ngrams()`.
`position`	Position represented by each occurrence: motif `"start"`, `"centre"`, or `"end"`.
`scale`	Position scale: one-based `"absolute"` sequence positions or `"relative"` positions from 0 to 1.
`by`	Optional character vector naming preserved metadata columns used to produce separate summaries, such as `"group"` or `"condition"`.

Details

Absolute positions use the one-based state index within each validated sequence. Relative positions are calculated as `(absolute_position - 1) / (n_states - 1)` and are constrained to the interval from 0 to 1. A sequence containing one state is assigned relative position 0.

Grouping is descriptive. The function does not test differences or attach behavioural, psychological, cognitive, or causal interpretations to motif location.

Value

A named list containing:

- summary: one row per motif and optional metadata group;
- occurrences: occurrence-level absolute, relative, and selected-scale positions;
- sequences, validation metadata, and extraction settings;
- settings: the resolved position basis, scale, and grouping columns.

The summary reports motif identifiers and labels, motif length, occurrence and sequence counts, and minimum, maximum, mean, and median positions.

Examples

```
sequences <- data.frame(
  id = c(rep("s1", 5L), rep("s2", 4L)),
  position = c(1:5, 1:4),
  state = c("A", "B", "A", "B", "C", "A", "B", "C", "B"),
  group = c(rep("g1", 5L), rep("g2", 4L))
)

extracted <- extract_sequence_ngrams(
  sequences,
  sequence_id_col = "id",
  order_col = "position",
  state_col = "state",
  metadata_cols = "group",
  min_length = 2,
  max_length = 3
)

positions <- summarise_sequence_motif_positions(
  extracted,
  position = "centre",
  scale = "relative",
  by = "group"
)

positions$summary
```

---

summarise_sequence_states

Summarise Sequence States

Description

Produces per-sequence and overall state-frequency summaries from validated ordered sequence data.

Usage

```
summarise_sequence_states(
  data,
  sequence_id_col,
  order_col,
  state_col,
  duration_col = NULL,
  metadata_cols = NULL,
  expected_states = NULL
)
```

Arguments

data	A data frame containing ordered state observations.
sequence_id_col	Name of the sequence identifier column.
order_col	Name of the numeric sequence-order column.
state_col	Name of the categorical state column.
duration_col	Optional name of a numeric duration column.
metadata_cols	Optional character vector naming columns that should remain constant within each sequence.
expected_states	Optional vector of known or permitted state values.

Details

Observation proportions use state rows as the denominator. Sequence proportions report the proportion of sequences in which each state occurs. Missing durations are excluded from duration calculations; an all-missing duration group returns NA.

Value

A named list containing:

- by_sequence: state counts and proportions within each sequence;
- overall: state counts and proportions across all sequences;
- audit, status, and mapping from input validation.

When duration_col is supplied, both tables also include duration sums, duration proportions, and mean durations.

Examples

```
sequences <- data.frame(
  id = c("s1", "s1", "s1", "s2", "s2"),
  position = c(1, 2, 3, 1, 2),
  state = c("A", "B", "A", "B", "C")
)
```

```
summaries <- summarise_sequence_states(
  sequences,
  sequence_id_col = "id",
  order_col = "position",
  state_col = "state"
)

summaries$by_sequence
summaries$overall
```

`summarise_sequence_transitions` *Summarise Adjacent Sequence Transitions*

Description

Counts transitions between adjacent ordered states for each sequence and across the complete data set.

Usage

```
summarise_sequence_transitions(
  data,
  sequence_id_col,
  order_col,
  state_col,
  metadata_cols = NULL,
  expected_states = NULL,
  include_self = TRUE
)
```

Arguments

`data`	A data frame containing ordered state observations.
`sequence_id_col`	Name of the sequence identifier column.
`order_col`	Name of the numeric sequence-order column.
`state_col`	Name of the categorical state column.
`metadata_cols`	Optional character vector naming columns that should remain constant within each sequence.
`expected_states`	Optional vector of known or permitted state values.
`include_self`	Logical value indicating whether transitions from a state to the same state should be included.

Details

A transition is defined only between adjacent rows after deterministic ordering by sequence identifier, sequence order, and original row. Sequences with one state contribute no transitions.

Value

A named list containing:

- `by_sequence`: transition counts, proportions within each sequence, and conditional proportions within each origin state;
- `overall`: transition counts, sequence coverage, global proportions, and conditional origin-state proportions;
- `audit`, `status`, `mapping`, and the resolved `include_self` setting.

Examples

```
sequences <- data.frame(
  id = c("s1", "s1", "s1", "s2", "s2"),
  position = c(1, 2, 3, 1, 2),
  state = c("A", "B", "C", "A", "C")
)

transitions <- summarise_sequence_transitions(
  sequences,
  sequence_id_col = "id",
  order_col = "position",
  state_col = "state"
)

transitions$by_sequence
transitions$overall
```

`validate_sequence_data`

Validate Long-Format Sequence Data

Description

Produces a compact validation result based on `audit_sequence_data()` without modifying the input.

Usage

```
validate_sequence_data(
  data,
  sequence_id_col,
  order_col,
```

```
  state_col,
  duration_col = NULL,
  metadata_cols = NULL,
  expected_states = NULL
)
```

Arguments

`data`	A data frame containing ordered state observations.
`sequence_id_col`	Name of the sequence identifier column.
`order_col`	Name of the numeric sequence-order column.
`state_col`	Name of the categorical state column.
`duration_col`	Optional name of a numeric duration column.
`metadata_cols`	Optional character vector naming columns that should remain constant within each sequence.
`expected_states`	Optional vector of known or permitted state values.

Value

A named list containing `valid`, `status`, issue counts, the complete `audit` table, the column `mapping`, row and sequence counts, and observed `state_levels`. A result is valid when no error-severity issue is present. Review-severity issues do not automatically invalidate the input.

Examples

```
sequences <- data.frame(
  id = rep(c("s1", "s2"), each = 2L),
  position = rep(1:2, times = 2L),
  state = c("home", "search", "home", "product")
)

validation <- validate_sequence_data(
  sequences,
  sequence_id_col = "id",
  order_col = "position",
  state_col = "state"
)

validation$status
validation$valid
```

Index