Package ‘survinger’

April 5, 2026

Title Design-Adjusted Inference for Pathogen Lineage Surveillance

Version 0.1.1

Description Provides tools for optimizing sequencing resource allocation and estimating pathogen lineage prevalence under real-world genomic surveillance conditions. Implements constrained allocation optimization for limited sequencing capacity across multiple regions and sample sources. Includes Horvitz-Thompson and post-stratified estimators that account for unequal sequencing rates, delay-adjusted nowcasting for right-censored reporting data, and combined design-weighted delay-corrected inference with uncertainty propagation.

License MIT + file LICENSE

URL https://github.com/CuiweiG/survinger

BugReports https://github.com/CuiweiG/survinger/issues

Depends R (>= 4.1.0)

Imports checkmate (>= 2.0.0), cli (>= 3.0.0), dplyr (>= 1.1.0), generics, ggplot2 (>= 3.4.0), methods, purrr, rlang (>= 1.0.0), stats, tibble

Suggests MASS, Matrix, nloptr (>= 2.0.0), survey, tidyr, vctrs, covr, knitr, rmarkdown, scales, testthat (>= 3.0.0), withr

Config/testthat/edition 3

Encoding UTF-8

Language en-US

LazyData true

RoxygenNote 7.3.3

VignetteBuilder knitr

NeedsCompilation no

Author Cuiwei Gao [aut, cre, cph]

Maintainer Cuiwei Gao <48gaocuiwei@gmail.com>

Repository CRAN

Date/Publication 2026-04-05 11:10:02 UTC

Contents

`glance.surv` *One-row summary of survinger model*

Description

One-row summary of survinger model

Usage

```
## S3 method for class 'surv_prevalence'
glance(x, ...)

## S3 method for class 'surv_delay_fit'
glance(x, ...)

## S3 method for class 'surv_adjusted'
glance(x, ...)
```

Arguments

`x`	A survinger result object.
`...`	Additional arguments (currently unused).

Value

A single-row tibble with key summary statistics.

Examples

```
sim <- surv_simulate(n_regions = 3, n_weeks = 10, seed = 1)
d <- surv_design(sim$sequences, ~ region,
                 sim$population[c("region", "seq_rate")], sim$population)
prev <- surv_lineage_prevalence(d, "BA.2.86")
glance(prev)
```

`plot.surv` *Plot methods for survinger objects*

Description

Visualize surveillance design, allocation, prevalence estimates, delay distributions, nowcasts, and adjusted estimates.

Usage

```
## S3 method for class 'surv_design'
plot(x, ...)

## S3 method for class 'surv_allocation'
plot(x, ...)

## S3 method for class 'surv_prevalence'
plot(x, ...)

## S3 method for class 'surv_delay_fit'
```

```
plot(x, ...)

## S3 method for class 'surv_nowcast'
plot(x, ...)

## S3 method for class 'surv_adjusted'
plot(x, ...)
```

Arguments

`x`	A survinger object.
`...`	Additional arguments (currently unused).

Value

A ggplot2 object.

`print.surv_adjusted` *Combined design-weighted and delay-adjusted prevalence*

Description

Simultaneously corrects for unequal sequencing rates and right-truncation from reporting delays.

Usage

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

## S3 method for class 'surv_adjusted'
as.data.frame(x, ...)

surv_adjusted_prevalence(
  design,
  delay_fit,
  lineage,
  time = "epiweek",
  prevalence_method = "hajek",
  nowcast_method = "direct",
  conf_level = 0.95,
  bootstrap_n = 0L
)
```

Arguments

`x`	Object to print.
`...`	Additional arguments (unused).
`design`	A `surv_design` object.
`delay_fit`	A `surv_delay_fit` object.
`lineage`	Character. Target lineage.
`time`	Character. Default `"epiweek"`.
`prevalence_method`	Character. Default `"hajek"`.
`nowcast_method`	Character. Default `"direct"`.
`conf_level`	Numeric. Default 0.95.
`bootstrap_n`	Integer. 0 for delta method, >0 for bootstrap. Default 0.

Value

Invisibly returns the input object.

A `surv_adjusted` object.

Examples

```
sim <- surv_simulate(n_regions = 3, n_weeks = 12, seed = 1)
d <- surv_design(sim$sequences, ~ region,
                 sim$population[c("region", "seq_rate")], sim$population)
delay <- surv_estimate_delay(d)
adj <- surv_adjusted_prevalence(d, delay, "BA.2.86")
print(adj)
```

`print.surv_allocation` *Optimize sequencing allocation across strata*

Description

Given fixed total sequencing capacity, finds the optimal allocation across strata that minimizes a specified objective function.

Usage

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

## S3 method for class 'surv_allocation'
as.data.frame(x, ...)
```

```
surv_optimize_allocation(
  design,
  objective = c("min_mse", "max_detection", "min_imbalance"),
  total_capacity,
  budget = NULL,
  min_per_stratum = 2L,
  target_lineage = NULL,
  target_prevalence = 0.01,
  cost_col = NULL
)
```

Arguments

`x`	Object to print.
`...`	Additional arguments (unused).
`design`	A `surv_design` object.
`objective`	Character. One of `"min_mse"`, `"max_detection"`, or `"min_imbalance"`.
`total_capacity`	Integer. Total sequences available.
`budget`	Numeric or `NULL`. Optional budget constraint.
`min_per_stratum`	Integer. Minimum per stratum. Default 2.
`target_lineage`	Character. Required for `"max_detection"`.
`target_prevalence`	Numeric. Assumed prevalence for detection. Default 0.01.
`cost_col`	Character or `NULL`. Column name for per-sequence cost.

Value

Invisibly returns the input object.

A `surv_allocation` object.

Examples

```
sim <- surv_simulate(n_regions = 4, n_weeks = 10, seed = 1)
d <- surv_design(sim$sequences, ~ region,
                 sim$population[c("region", "seq_rate")], sim$population)
a <- surv_optimize_allocation(d, "min_mse", total_capacity = 500)
print(a)
```

`print.surv_delay_fit` *Estimate reporting delay distribution*

Description

Fits a parametric distribution to the delay between sample collection and sequence reporting, accounting for right-truncation.

Usage

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

surv_estimate_delay(
  design,
  distribution = c("negbin", "poisson", "lognormal", "nonparametric"),
  strata = NULL,
  max_delay = 60L,
  ref_date = NULL
)
```

Arguments

`x`	Object to print.
`...`	Additional arguments (unused).
`design`	A `surv_design` object with both collection and report dates.
`distribution`	Character: `"negbin"` (default), `"poisson"`, `"lognormal"`, or `"nonparametric"`.
`strata`	Optional one-sided formula for delay stratification.
`max_delay`	Integer. Maximum plausible delay in days. Default 60.
`ref_date`	Date. Reference date for right-truncation. Default is max report date in data.

Value

Invisibly returns the input object.

A `surv_delay_fit` object.

Examples

```
sim <- surv_simulate(n_regions = 3, n_weeks = 12, seed = 1)
d <- surv_design(sim$sequences, ~ region,
                 sim$population[c("region", "seq_rate")], sim$population)
fit <- surv_estimate_delay(d)
print(fit)
```

`print.surv_design` *Create a genomic surveillance design object*

Description

Constructs a survey design object tailored for pathogen genomic surveillance, encoding stratification structure, sequencing rates, and population information needed for design-weighted inference.

Usage

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

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

## S3 method for class 'summary.surv_design'
print(x, ...)

surv_design(
  data,
  strata,
  sequencing_rate,
  population,
  date_collected = "collection_date",
  date_reported = "report_date",
  lineage = "lineage",
  source_type = NULL,
  source_config = NULL
)
```

Arguments

`x`	Object to print or summarize.
`...`	Additional arguments (unused).
`object`	A `surv_design` object to summarize.
`data`	Data frame of individual sequence records.
`strata`	One-sided formula specifying stratification variables (e.g., ~ `region` or ~ `region + source_type`).
`sequencing_rate`	Either a one-sided formula (~ `n_sequenced / n_positive`), a named numeric vector, or a data frame with strata columns and `seq_rate`.
`population`	Data frame with one row per stratum, containing stratification variables and population-level denominators.
`date_collected`	Column name for collection date. Default `"collection_date"`.

`date_reported`	Column name for report date. Default `"report_date"`. Set `NULL` if unavailable.
`lineage`	Column name for lineage. Default `"lineage"`.
`source_type`	Column name for sample source. Default `NULL`.
`source_config`	Optional tibble of per-source characteristics.

Value

Invisibly returns the input object.

A summary list of class `summary.surv_design`.

Invisibly returns the input object.

An object of class `surv_design`.

See Also

`surv_simulate()`, `surv_lineage_prevalence()`, `surv_optimize_allocation()`

Examples

```
sim <- surv_simulate(n_regions = 3, n_weeks = 8, seed = 42)
design <- surv_design(
  data = sim$sequences,
  strata = ~ region,
  sequencing_rate = sim$population[c("region", "seq_rate")],
  population = sim$population
)
print(design)
```

`print.surv_nowcast` *Nowcast lineage counts correcting for reporting delays*

Description

Nowcast lineage counts correcting for reporting delays

Usage

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

## S3 method for class 'surv_nowcast'
as.data.frame(x, ...)

surv_nowcast_lineage(
  design,
  delay_fit,
```

```
  lineage = NULL,
  time = "epiweek",
  horizon = 4L,
  ref_date = NULL,
  method = c("direct", "em")
)
```

Arguments

`x`	Object to print.
`...`	Additional arguments (unused).
`design`	A `surv_design` object.
`delay_fit`	A `surv_delay_fit` object.
`lineage`	Character or `NULL`. Target lineage.
`time`	Character. Default `"epiweek"`.
`horizon`	Integer. Recent periods to nowcast. Default 4.
`ref_date`	Date or `NULL`. Default max report date.
`method`	Character: `"direct"` (default) or `"em"`.

Value

Invisibly returns the input object.

A `surv_nowcast` object.

Examples

```
sim <- surv_simulate(n_regions = 3, n_weeks = 12, seed = 1)
d <- surv_design(sim$sequences, ~ region,
                 sim$population[c("region", "seq_rate")], sim$population)
fit <- surv_estimate_delay(d)
nc <- surv_nowcast_lineage(d, fit, "BA.2.86")
print(nc)
```

`print.surv_prevalence` *Estimate lineage prevalence with design weights*

Description

Estimates the prevalence of a specified pathogen lineage over time, correcting for unequal sequencing rates across strata.

Usage

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

## S3 method for class 'surv_prevalence'
as.data.frame(x, ...)

surv_lineage_prevalence(
  design,
  lineage,
  time = "epiweek",
  method = c("hajek", "horvitz_thompson", "poststratified"),
  conf_level = 0.95,
  min_obs = 5L
)
```

Arguments

`x`	Object to convert.
`...`	Additional arguments (unused).
`design`	A `surv_design` object.
`lineage`	Character. Target lineage name.
`time`	Character. Time aggregation: `"epiweek"`, `"month"`, `"date"`, or a column name. Default `"epiweek"`.
`method`	Character. Estimation method: `"hajek"` (default), `"horvitz_thompson"`, or `"poststratified"`.
`conf_level`	Numeric. Confidence level. Default 0.95.
`min_obs`	Integer. Minimum observations per time period. Default 5.

Value

Invisibly returns the input object.

A data.frame.

A `surv_prevalence` object.

Examples

```
sim <- surv_simulate(n_regions = 3, n_weeks = 10, seed = 1)
d <- surv_design(sim$sequences, ~ region,
                 sim$population[c("region", "seq_rate")], sim$population)
prev <- surv_lineage_prevalence(d, "BA.2.86")
print(prev)
```

`sarscov2_surveillance` *Example SARS-CoV-2 genomic surveillance data*

Description

Simulated genomic surveillance dataset with 5 regions, 26 weeks, highly unequal sequencing rates (15% to 0.5%), three sample sources, and negative binomial reporting delays. Contains known ground truth for benchmarking.

Usage

```
sarscov2_surveillance
```

Format

A named list with four elements:

- **sequences** Tibble of sequence records: sequence_id, region, source_type, lineage, collection_date, report_date, epiweek.
- **population** Tibble with one row per region: region, n_positive, n_sequenced, seq_rate, pop_total.
- **truth** Tibble of true lineage prevalence by region and week.
- **parameters** List of simulation parameters.

Source

Simulated using `surv_simulate(seed = 20240101)`.

Examples

```
data(sarscov2_surveillance)
head(sarscov2_surveillance$sequences)
sarscov2_surveillance$population
```

`surv_bind` *Combine multiple prevalence estimates*

Description

Bind results from different lineages or methods into a single tibble for comparison plots and tables.

Usage

```
surv_bind(...)
```

Arguments

`...`	`surv_prevalence` objects to combine.

Value

A tibble with a `source` column identifying each input.

See Also

`surv_lineage_prevalence()`, `surv_sensitivity()`

Examples

```
sim <- surv_simulate(n_regions = 3, n_weeks = 10, seed = 1)
d <- surv_design(sim$sequences, ~ region,
                 sim$population[c("region", "seq_rate")], sim$population)
p1 <- surv_lineage_prevalence(d, "BA.5")
p2 <- surv_lineage_prevalence(d, "XBB.1.5")
combined <- surv_bind(p1, p2)
head(combined)
```

`surv_compare_allocations` *Compare multiple allocation strategies*

Description

Compare multiple allocation strategies

Usage

```
surv_compare_allocations(
  design,
 strategies = c("equal", "proportional", "min_mse", "max_detection", "min_imbalance"),
  total_capacity,
  target_prevalence = 0.01,
  ...
)
```

Arguments

`design`	A `surv_design` object.
`strategies`	Character vector. Default includes all built-in.
`total_capacity`	Integer. Total sequences.
`target_prevalence`	Numeric. For detection objective.
`...`	Passed to `surv_optimize_allocation()`.

Value

A tibble comparing strategies.

Examples

```
sim <- surv_simulate(n_regions = 4, n_weeks = 10, seed = 1)
d <- surv_design(sim$sequences, ~ region,
                 sim$population[c("region", "seq_rate")], sim$population)
surv_compare_allocations(d, total_capacity = 200)
```

`surv_compare_estimates`

Compare weighted vs naive prevalence estimates

Description

Side-by-side plot showing the impact of design correction.

Usage

```
surv_compare_estimates(weighted, naive, title = NULL)
```

Arguments

`weighted`	A `surv_prevalence` object (design-weighted).
`naive`	A `surv_prevalence` object (unweighted).
`title`	Character or `NULL`. Plot title.

Value

A ggplot2 object.

Examples

```
sim <- surv_simulate(n_regions = 3, n_weeks = 10, seed = 1)
d <- surv_design(sim$sequences, ~ region,
                 sim$population[c("region", "seq_rate")], sim$population)
w <- surv_lineage_prevalence(d, "BA.2.86")
n <- surv_naive_prevalence(d, "BA.2.86")
surv_compare_estimates(w, n)
```

`surv_design_effect` *Compute design effect over time*

Description

Compute design effect over time

Usage

```
surv_design_effect(weighted, naive)
```

Arguments

`weighted`	A `surv_prevalence` object.
`naive`	A `surv_prevalence` object.

Value

A tibble with time, deff, and bias_correction columns.

Examples

```
sim <- surv_simulate(n_regions = 3, n_weeks = 10, seed = 1)
d <- surv_design(sim$sequences, ~ region,
                 sim$population[c("region", "seq_rate")], sim$population)
w <- surv_lineage_prevalence(d, "BA.2.86")
n <- surv_naive_prevalence(d, "BA.2.86")
surv_design_effect(w, n)
```

`surv_detection_probability` *Variant detection probability under current design*

Description

Variant detection probability under current design

Usage

```
surv_detection_probability(
  design,
  true_prevalence,
  delay_fit = NULL,
  n_periods = 1L,
  detection_threshold = 1L
)
```

Arguments

`design`	A `surv_design` object.
`true_prevalence`	Numeric in (0,1).
`delay_fit`	Optional `surv_delay_fit`.
`n_periods`	Integer. Accumulation periods. Default 1.
`detection_threshold`	Integer. Min detections. Default 1.

Value

A list with `overall`, `cumulative`, `by_stratum`, `parameters`.

Examples

```
sim <- surv_simulate(n_regions = 3, n_weeks = 10, seed = 1)
d <- surv_design(sim$sequences, ~ region,
                 sim$population[c("region", "seq_rate")], sim$population)
surv_detection_probability(d, 0.01)
```

`surv_estimate` *Pipe-friendly surveillance analysis*

Description

Convenience wrapper that creates a design, estimates prevalence, and optionally applies delay correction in a single pipe-friendly call. Designed for rapid exploratory analysis in interactive sessions.

Usage

```
surv_estimate(
  data,
  strata,
  sequencing_rate,
  population,
  lineage,
  correct_delay = FALSE,
  method = "hajek",
  ...
)
```

Arguments

`data`	Data frame of sequence records.
`strata`	One-sided formula for stratification.
`sequencing_rate`	Sequencing rate specification (see `surv_design()`).
`population`	Population data frame.
`lineage`	Character. Target lineage to estimate.
`correct_delay`	Logical. Apply delay correction? Default `FALSE`.
`method`	Character. Prevalence method. Default `"hajek"`.
`...`	Additional arguments passed to `surv_design()`.

Value

A `surv_prevalence` or `surv_adjusted` object.

See Also

`surv_design()`, `surv_lineage_prevalence()`

Examples

```
sim <- surv_simulate(n_regions = 3, n_weeks = 10, seed = 1)
# One-liner analysis:
result <- surv_estimate(
  data = sim$sequences, strata = ~ region,
  sequencing_rate = sim$population[c("region", "seq_rate")],
  population = sim$population,
  lineage = "BA.2.86"
)
print(result)
```

`surv_filter` *Subset a surveillance design by filter criteria*

Description

Creates a new `surv_design` object containing only sequences matching the specified filter criteria.

Usage

```
surv_filter(design, ...)
```

Arguments

`design`	A `surv_design` object.
`...`	Filter expressions passed to `dplyr::filter()`.

Value

A new `surv_design` object with filtered data.

See Also

`surv_design()`

Examples

```
sim <- surv_simulate(n_regions = 5, n_weeks = 12, seed = 1)
d <- surv_design(sim$sequences, ~ region,
                 sim$population[c("region", "seq_rate")], sim$population)
d_sub <- surv_filter(d, region %in% c("Region_A", "Region_B"))
print(d_sub)
```

`surv_naive_prevalence` *Compute naive (unweighted) lineage prevalence*

Description

Simple proportion without design correction. Useful as baseline.

Usage

```
surv_naive_prevalence(design, lineage, time = "epiweek", conf_level = 0.95)
```

Arguments

`design`	A `surv_design` object.
`lineage`	Character. Target lineage.
`time`	Character. Time aggregation. Default `"epiweek"`.
`conf_level`	Numeric. Default 0.95.

Value

A `surv_prevalence` object with `method` = `"naive"`.

Examples

```
sim <- surv_simulate(n_regions = 3, n_weeks = 10, seed = 1)
d <- surv_design(sim$sequences, ~ region,
                 sim$population[c("region", "seq_rate")], sim$population)
naive <- surv_naive_prevalence(d, "BA.2.86")
```

`surv_plot_allocation` *Plot allocation plan*

Description

Plot allocation plan

Usage

```
surv_plot_allocation(allocation)
```

Arguments

`allocation`	A `surv_allocation` object.

Value

A ggplot2 object.

Examples

```
sim <- surv_simulate(n_regions = 5, n_weeks = 10, seed = 1)
d <- surv_design(sim$sequences, ~ region,
                 sim$population[c("region", "seq_rate")], sim$population)
a <- surv_optimize_allocation(d, "min_mse", total_capacity = 500)
surv_plot_allocation(a)
```

`surv_plot_sequencing_rates` *Plot sequencing rate inequality across strata*

Description

Plot sequencing rate inequality across strata

Usage

```
surv_plot_sequencing_rates(design)
```

Arguments

`design`	A `surv_design` object.

Value

A ggplot2 object.

Examples

```
sim <- surv_simulate(n_regions = 5, n_weeks = 10, seed = 1)
d <- surv_design(sim$sequences, ~ region,
                 sim$population[c("region", "seq_rate")], sim$population)
surv_plot_sequencing_rates(d)
```

`surv_power_curve` *Compute power curve for detection across prevalence range*

Description

Generates a detection probability curve that can be directly plotted or included in publications. Answers: "At what prevalence does our surveillance achieve X% detection?"

Usage

```
surv_power_curve(
  design,
  prevalence_range = seq(0.001, 0.05, by = 0.001),
  delay_fit = NULL,
  thresholds = c(0.5, 0.8, 0.95)
)

## S3 method for class 'surv_power_curve'
plot(x, ...)
```

Arguments

`design`	A `surv_design` object.
`prevalence_range`	Numeric vector of prevalences to evaluate. Default `seq(0.001, 0.05, by = 0.001)`.
`delay_fit`	Optional `surv_delay_fit`.
`thresholds`	Numeric vector of detection thresholds to mark. Default `c(0.5, 0.8, 0.95)`.
`x`	A `surv_power_curve` object.
`...`	Additional arguments (unused).

Value

A list with:

curve Tibble with prevalence and detection columns.

thresholds Tibble with threshold, prevalence_needed columns.

A ggplot2 object.

See Also

`surv_detection_probability()`, `surv_required_sequences()`

Examples

```
sim <- surv_simulate(n_regions = 3, n_weeks = 10, seed = 1)
d <- surv_design(sim$sequences, ~ region,
                 sim$population[c("region", "seq_rate")], sim$population)
pc <- surv_power_curve(d)
pc$thresholds
```

`surv_prevalence_by` *Estimate prevalence by subgroup*

Description

Applies `surv_lineage_prevalence()` within subgroups defined by a grouping variable. Analogous to `survey::svyby()` for stratified survey analysis.

Usage

```
surv_prevalence_by(
  design,
  lineage,
  by,
  time = "epiweek",
  method = "hajek",
  conf_level = 0.95
)
```

Arguments

`design`	A `surv_design` object.
`lineage`	Character. Target lineage.
`by`	Character. Column name to group by (e.g., `"region"` or `"source_type"`).
`time`	Character. Time aggregation. Default `"epiweek"`.
`method`	Character. Estimation method. Default `"hajek"`.
`conf_level`	Numeric. Default 0.95.

Value

A tibble with columns: group, time, lineage, prevalence, se, ci_lower, ci_upper, n_obs, effective_n.

See Also

`surv_lineage_prevalence()`, `surv_filter()`

Examples

```
sim <- surv_simulate(n_regions = 4, n_weeks = 10, seed = 1)
d <- surv_design(sim$sequences, ~ region,
                 sim$population[c("region", "seq_rate")],
                 sim$population, source_type = "source_type")
surv_prevalence_by(d, "BA.2.86", by = "region")
```

`surv_quality` *Compute surveillance quality metrics*

Description

Returns a single-row tibble of design quality indicators suitable for inclusion in manuscripts. Analogous to `broom::glance()` but for the surveillance design itself.

Usage

```
surv_quality(design, target_lineage = NULL, target_prevalence = 0.01)
```

Arguments

`design`	A `surv_design` object.
`target_lineage`	Character or `NULL`. Default `NULL` auto-selects.
`target_prevalence`	Numeric. For detection calculation. Default 0.01.

Value

A single-row tibble with columns: n_obs, n_strata, gini, rate_ratio, effective_n, deff, detection_prob, mean_bias.

See Also

`surv_report()`, `surv_design()`

Examples

```
sim <- surv_simulate(n_regions = 3, n_weeks = 10, seed = 1)
d <- surv_design(sim$sequences, ~ region,
                 sim$population[c("region", "seq_rate")], sim$population)
surv_quality(d)
```

`surv_report` *Generate a comprehensive surveillance system report*

Description

Produces a summary of the current surveillance design's strengths, weaknesses, and recommendations.

Usage

```
surv_report(design, target_lineage = NULL, target_prevalence = 0.01)
```

Arguments

`design`	A `surv_design` object.
`target_lineage`	Character or `NULL`. Lineage to focus on. If `NULL`, uses the most common non-"Other" lineage.
`target_prevalence`	Numeric. Assumed prevalence for detection calculations. Default 0.01.

Value

Invisibly returns a named list of computed metrics including n_obs, n_strata, rate_range, gini, effective_n, detection_prob, and mean_bias.

See Also

`surv_design()`, `surv_lineage_prevalence()`, `surv_detection_probability()`, `surv_optimize_allocation()`

Examples

```
sim <- surv_simulate(n_regions = 3, n_weeks = 10, seed = 1)
d <- surv_design(sim$sequences, ~ region,
                 sim$population[c("region", "seq_rate")], sim$population)
surv_report(d)
```

`surv_reporting_probability` *Compute cumulative reporting probability*

Description

Compute cumulative reporting probability

Usage

```
surv_reporting_probability(delay_fit, delta, stratum = NULL)
```

Arguments

`delay_fit`	A `surv_delay_fit` object.
`delta`	Integer vector. Days since collection.
`stratum`	Character or `NULL`. Default `NULL`.

Value

Numeric vector of probabilities.

Examples

```
sim <- surv_simulate(n_regions = 3, n_weeks = 12, seed = 1)
d <- surv_design(sim$sequences, ~ region,
                 sim$population[c("region", "seq_rate")], sim$population)
fit <- surv_estimate_delay(d)
surv_reporting_probability(fit, delta = c(7, 14, 21))
```

`surv_required_sequences` *Required sequences for target detection probability*

Description

Required sequences for target detection probability

Usage

```
surv_required_sequences(
  true_prevalence,
  target_probability = 0.95,
  n_periods = 1L,
  detection_threshold = 1L
)
```

Arguments

`true_prevalence`	Numeric.
`target_probability`	Numeric. Default 0.95.
`n_periods`	Integer. Default 1.
`detection_threshold`	Integer. Default 1.

Value

Integer.

Examples

```
surv_required_sequences(0.01)
surv_required_sequences(0.05, target_probability = 0.99)
```

`surv_sensitivity` *Sensitivity analysis across methods*

Description

Runs all three prevalence estimators and delay/no-delay variants on the same design, producing a comparison table. Essential for robustness checks in publications.

Usage

```
surv_sensitivity(
  design,
  lineage,
  delay_fit = NULL,
  time = "epiweek",
  conf_level = 0.95
)
```

Arguments

`design`	A `surv_design` object.
`lineage`	Character. Target lineage.
`delay_fit`	Optional `surv_delay_fit` object. If provided, includes delay-adjusted estimates.
`time`	Character. Default `"epiweek"`.
`conf_level`	Numeric. Default 0.95.

Value

A tibble with one row per method-time combination, columns: method, time, prevalence, se, ci_lower, ci_upper.

See Also

`surv_lineage_prevalence()`, `surv_adjusted_prevalence()`

Examples

```
sim <- surv_simulate(n_regions = 3, n_weeks = 10, seed = 1)
d <- surv_design(sim$sequences, ~ region,
                 sim$population[c("region", "seq_rate")], sim$population)
surv_sensitivity(d, "BA.2.86")
```

`surv_set_weights` *Override design weights with custom values*

Description

Override design weights with custom values

Usage

```
surv_set_weights(design, weights)
```

Arguments

`design`	A `surv_design` object.
`weights`	Numeric vector of length equal to number of strata.

Value

Updated `surv_design`.

Examples

```
sim <- surv_simulate(n_regions = 3, n_weeks = 4, seed = 1)
d <- surv_design(sim$sequences, ~ region,
                 sim$population[c("region", "seq_rate")], sim$population)
d2 <- surv_set_weights(d, rep(1.0, d$n_strata))
```

`surv_simulate` *Simulate genomic surveillance data*

Description

Generates synthetic surveillance datasets with realistic features: multiple regions with unequal sequencing rates, multiple lineages with time-varying prevalence, configurable reporting delays, and multiple sample sources.

Usage

```
surv_simulate(
  n_regions = 5L,
  n_weeks = 26L,
  total_positive_per_week = 1000L,
  sequencing_rates = NULL,
  lineage_dynamics = NULL,
  delay_params = list(mu = 10, size = 3),
  sources = c("clinical", "wastewater", "sentinel"),
  source_weights = c(0.7, 0.2, 0.1),
  seed = NULL
)
```

Arguments

`n_regions`	Integer. Number of geographic regions. Default 5.
`n_weeks`	Integer. Number of epiweeks. Default 26.
`total_positive_per_week`	Integer. Mean total positive cases per week across all regions. Default 1000.
`sequencing_rates`	Numeric vector of length `n_regions`. Per-region sequencing probability. If `NULL`, generated from a Beta distribution with realistic inequality. Default `NULL`.
`lineage_dynamics`	Named list of functions, each taking a week number and returning a positive weight. If `NULL`, uses a default four-lineage scenario. Default `NULL`.
`delay_params`	List with `mu` and `size` for negative binomial reporting delay. Default `list(mu = 10, size = 3)`.
`sources`	Character vector of sample source types. Default `c("clinical", "wastewater", "sentinel")`.
`source_weights`	Numeric vector (same length as `sources`). Default `c(0.7, 0.2, 0.1)`.
`seed`	Integer or `NULL`. Random seed. Default `NULL`.

Value

A named list with elements:

sequences	Tibble of individual sequence records.
population	Tibble with one row per region.
truth	Tibble of true lineage prevalence by region and week.
parameters	List of all input parameters.

Examples

```
sim <- surv_simulate(n_regions = 3, n_weeks = 8, seed = 42)
head(sim$sequences)
sim$population
```

`surv_table` *Format prevalence results for knitr tables*

Description

Produces a publication-ready summary table from any survinger result. Automatically called when objects are printed inside RMarkdown chunks if `knitr` is loaded.

Usage

```
surv_table(x, digits = 3, percent = TRUE, ...)
```

Arguments

`x`	A survinger result object.
`digits`	Integer. Decimal places for prevalence. Default 3.
`percent`	Logical. Display as percentages? Default `TRUE`.
`...`	Additional arguments (unused).

Value

A tibble formatted for display.

See Also

`surv_lineage_prevalence()`, `tidy.surv_prevalence()`

Examples

```
sim <- surv_simulate(n_regions = 3, n_weeks = 10, seed = 1)
d <- surv_design(sim$sequences, ~ region,
                 sim$population[c("region", "seq_rate")], sim$population)
prev <- surv_lineage_prevalence(d, "BA.2.86")
surv_table(prev)
```

`surv_update_rates` *Update sequencing rates in a surveillance design*

Description

Update sequencing rates in a surveillance design

Usage

```
surv_update_rates(design, new_rates)
```

Arguments

`design`	A `surv_design` object.
`new_rates`	Data frame with strata columns + `seq_rate`, or named numeric vector.

Value

Updated `surv_design`.

Examples

```
sim <- surv_simulate(n_regions = 3, n_weeks = 4, seed = 1)
d <- surv_design(sim$sequences, ~ region,
                 sim$population[c("region", "seq_rate")], sim$population)
new_r <- sim$population[c("region", "seq_rate")]
new_r$seq_rate <- new_r$seq_rate * 2
d2 <- surv_update_rates(d, new_r)
```

`theme_survinger` *Publication-quality ggplot2 theme*

Description

A clean, high-contrast theme suitable for journal publication. Follows Nature/Lancet figure guidelines.

Usage

```
theme_survinger(base_size = 11, base_family = "")
```

Arguments

`base_size`	Base font size. Default 11.
`base_family`	Font family. Default `""`.

Value

A ggplot2 theme object.

See Also

`surv_compare_estimates()`

Examples

```
library(ggplot2)
ggplot(mtcars, aes(wt, mpg)) + geom_point() + theme_survinger()
```

`tidy.surv` *Extract tidy estimates from survinger objects*

Description

Converts survinger result objects into tidy tibbles suitable for further analysis with dplyr, ggplot2, or other tidyverse tools.

Usage

```
## S3 method for class 'surv_prevalence'
tidy(x, ...)

## S3 method for class 'surv_nowcast'
tidy(x, ...)

## S3 method for class 'surv_adjusted'
tidy(x, ...)

## S3 method for class 'surv_allocation'
tidy(x, ...)

## S3 method for class 'surv_delay_fit'
tidy(x, ...)
```

Arguments

`x`	A survinger result object.
`...`	Additional arguments (currently unused).

Value

A tibble.

Examples

```
sim <- surv_simulate(n_regions = 3, n_weeks = 10, seed = 1)
d <- surv_design(sim$sequences, ~ region,
                 sim$population[c("region", "seq_rate")], sim$population)
prev <- surv_lineage_prevalence(d, "BA.2.86")
tidy(prev)
```

Index