build_phi_function_from_coefs

convert a pair of simple logistic regression coefficients into $P(Y|T)$ curve:

Description

convert a pair of simple logistic regression coefficients into $P(Y|T)$ curve:

Usage

build_phi_function_from_coefs(coefs)

Arguments

coops numeric vector of coefficients

Value

function(t) $P(Y=1|T=t)$

compute_mu

compute mean window period duration from simple logistic regression coefficients

Description

compute mean window period duration from simple logistic regression coefficients

Usage

compute_mu(theta)

Arguments

theta numeric vector of coefficients
Value

numeric scalar: mean window period duration

**Description**

This function fits a logistic regression model for a binary outcome Y with an interval-censored covariate T, using an EM algorithm, as described in Morrison et al (2021); doi: 10.1111/biom.13472.

**Usage**

```r
fit_joint_model(
  participant_level_data,
  obs_level_data,
  model_formula = stats::formula(Y ~ T),
  mu_function = compute_mu,
  bin_width = 1,
  denom_offset = 0.1,
  EM_toler_loglik = 0.1,
  EM_toler_est = 1e-04,
  EM_max_iterations = Inf,
  glm_tolerance = 1e-07,
  glm_maxit = 20,
  initial_S_estimate_location = 0.25,
  coef_change_metric = "max abs rel diff coefs",
  verbose = FALSE
)
```

**Arguments**

- `participant_level_data` a data.frame or tibble with the following variables:
  - ID: participant ID
  - E: study enrollment date
  - L: date of last negative test for seroconversion
  - R: date of first positive test for seroconversion
  - Cohort (optional): this variable can be used to stratify the modeling of the seroconversion distribution.

- `obs_level_data` a data.frame or tibble with the following variables:
  - ID: participant ID
  - O: biomarker sample collection dates
  - Y: MAA classifications (binary outcomes)

- `model_formula` the functional form for the regression model for p(y|t) (as a formula() object)
**mu_function**
a function taking a vector of regression coefficient estimates as input and outputting an estimate of mu (mean duration of MAA-positive infection).

**bin_width**
the number of days between possible seroconversion dates (should be an integer)

**denom_offset**
an offset value added to the denominator of the hazard estimates to improve numerical stability

**EM_toler_loglik**
the convergence cutoff for the log-likelihood criterion ("Delta_L" in the paper)

**EM_toler_est**
the convergence cutoff for the parameter estimate criterion ("Delta_theta" in the paper)

**EM_max_iterations**
the number of EM iterations to perform before giving up if still not converged.

**glm_tolerance**
the convergence cutoff for the glm fit in the M step

**glm_maxit**
the iterations cutoff for the glm fit in the M step

**initial_S_estimate_location**
determines how seroconversion date is guessed to initialize the algorithm; can be any decimal between 0 and 1; 0.5 = midpoint imputation, 0.25 = 1st quartile, 0 = last negative, etc.

**coef_change_metric**
a string indicating the type of parameter estimate criterion to use:

- "max abs rel diff coefs" is the "Delta_theta" criterion described in the paper.
- "max abs diff coefs" is the maximum absolute change in the coefficients (not divided by the old values); this criterion can be useful when some parameters are close to 0.
- "diff mu" is the absolute change in mu, which may be helpful in the incidence estimate calibration setting but not elsewhere.

**verbose**
whether to print algorithm progress details to the console

**Value**
a list with the following elements:

- **Theta**: the estimated regression coefficients for the model of \( p(Y|T) \)
- **Mu**: the estimated mean window period (a transformation of Theta)
- **Omega**: a table with the estimated parameters for the model of \( p(S|E) \).
- **converged**: indicator of whether the algorithm reached its cutoff criteria before reaching the specified maximum iterations. 1 = reached cutoffs, 0 = not.
- **iterations**: the number of EM iterations completed before the algorithm stopped.
- **convergence_metrics**: the four convergence metrics

**References**

Examples

## Not run:

```r
# simulate data:
study_data <- simulate_interval_censoring()

# fit model:
EM_algorithm_outputs <- fit_joint_model(
  obs_level_data = study_data$obs_data,
  participant_level_data = study_data$pt_data
)

## End(Not run)
```

### fit_midpoint_model

**Fit model using midpoint imputation**

#### Description

Fit model using midpoint imputation

#### Usage

```r
fit_midpoint_model(
  participant_level_data, 
  obs_level_data, 
  maxit = 1000, 
  tolerance = 1e-08
)
```

#### Arguments

- `participant_level_data`: a data.frame or tibble with the following variables:
  - ID: participant ID
  - E: study enrollment date
  - L: date of last negative test for seroconversion
  - R: date of first positive test for seroconversion
  - `Cohort` (optional): this variable can be used to stratify the modeling of the seroconversion distribution.

- `obs_level_data`: a data.frame or tibble with the following variables:
  - ID: participant ID
  - O: biomarker sample collection dates
  - Y: MAA classifications (binary outcomes)

- `maxit`: maximum iterations, passed to `bigglm`

- `tolerance`: convergence criterion, passed to `bigglm`
**fit_uniform_model**

Fit model using uniform imputation

**Usage**

```r
fit_uniform_model(
  participant_level_data,
  obs_level_data,
  maxit = 1000,
  tolerance = 1e-08,
  n_imputations = 10
)
```

**Arguments**

- `participant_level_data`: a data.frame or tibble with the following variables:
  - ID: participant ID
  - E: study enrollment date
  - L: date of last negative test for seroconversion
  - R: date of first positive test for seroconversion
  - Cohort (optional): this variable can be used to stratify the modeling of the seroconversion distribution.

- `obs_level_data`: a data.frame or tibble with the following variables:

**Value**

a vector of logistic regression coefficient estimates

**Examples**

```r
sim_data = simulate_interval_censoring(
  "theta" = c(0.986, -3.88),
  "study_cohort_size" = 4500,
  "preconversion_interval_length" = 365,
  "hazard_alpha" = 1,
  "hazard_beta" = 0.5)

theta_est_midpoint = fit_midpoint_model(
  obs_level_data = sim_data$obs_data,
  participant_level_data = sim_data$pt_data
)
```
plot_CDF

- ID: participant ID
- O: biomarker sample collection dates
- Y: MAA classifications (binary outcomes)

maxit maximum iterations, passed to bigglm
tolerance convergence criterion, passed to bigglm
n_imputations number of imputed data sets to create

Value

a vector of logistic regression coefficient estimates

Examples

```r
sim_data = simulate_interval_censoring(
  "theta" = c(0.986, -3.88),
  "study_cohort_size" = 4500,
  "preconversion_interval_length" = 365,
  "hazard_alpha" = 1,
  "hazard_beta" = 0.5)

theta_est_midpoint = fit_uniform_model(
  obs_level_data = sim_data$obs_data,
  participant_level_data = sim_data$pt_data)
```

plot_CDF

plot estimated and true CDFs for seroconversion date distribution

Description

plot estimated and true CDFs for seroconversion date distribution

Usage

```r
plot_CDF(true_hazard_alpha, true_hazard_beta, omega.hat)
```

Arguments

- `true_hazard_alpha` The data-generating hazard at the start of the study
- `true_hazard_beta` The change in data-generating hazard per calendar year
- `omega.hat` tibble of estimated discrete hazards

Value

a ggplot
Examples

## Not run:

```r
hazard_alpha = 1
hazard_beta = 0.5
study_data <- simulate_interval_censoring(
    "hazard_alpha" = hazard_alpha,
    "hazard_beta" = hazard_beta)

# fit model:
EM_algorithm_outputs <- fit_joint_model(
    obs_level_data = study_data$obs_data,
    participant_level_data = study_data$pt_data
)
plot1 = plot_CDF(
    true_hazard_alpha = hazard_alpha,
    true_hazard_beta = hazard_beta,
    omega.hat = EM_algorithm_outputs$Omega)

print(plot1)

## End(Not run)
```

---

**plot_phi_curves**

*Plot true and estimated curves for P(Y=1|T=t)*

Description

Plot true and estimated curves for P(Y=1|T=t)

Usage

```r
plot_phi_curves(
    theta_true,
    theta.hat_joint,
    theta.hat_midpoint,
    theta.hat_uniform
)
```

Arguments

- `theta_true` the coefficients of the data-generating model P(Y=1|T=t)
- `theta.hat_joint` the estimated coefficients from the joint model
- `theta.hat_midpoint` the estimated coefficients from midpoint imputation
- `theta.hat_uniform` the estimated coefficients from uniform imputation
Value

a ggplot

Examples

```r
## Not run:
theta_true = c(0.986, -3.88)
hazard_alpha = 1
hazard_beta = 0.5
sim_data = simulate_interval_censoring(
  "theta" = theta_true,
  "study_cohort_size" = 4500,
  "preconversion_interval_length" = 365,
  "hazard_alpha" = hazard_alpha,
  "hazard_beta" = hazard_beta)

# extract the participant-level and observation-level simulated data:
sim_participant_data = sim_data$pt_data
sim_obs_data = sim_data$obs_data
rm(sim_data)

# joint model:
EM_algorithm_outputs = fit_joint_model(
  obs_level_data = sim_obs_data,
  participant_level_data = sim_participant_data,
  bin_width = 7,
  verbose = FALSE)

# midpoint imputation:
theta_est_midpoint = fit_midpoint_model(
  obs_level_data = sim_obs_data,
  participant_level_data = sim_participant_data
)

# uniform imputation:
theta_est_uniform = fit_uniform_model(
  obs_level_data = sim_obs_data,
  participant_level_data = sim_participant_data
)

plot2 = plot_phi_curves(
  theta_true = theta_true,
  theta.hat_uniform = theta_est_uniform,
  theta.hat_midpoint = theta_est_midpoint,
  theta.hat_joint = EM_algorithm_outputs$Theta)

print(plot2)

## End(Not run)
```
The `rwicc` package implements a regression model with an interval-censored covariate using an EM algorithm, as described in Morrison et al (2021); doi: 10.1111/biom.13472.

**rwicc functions**

The main `rwicc` functions are:

- `simulate_interval_censoring`
- `fit_joint_model`

**References**


---

The `seroconversion_inverse_survival_function` determines the seroconversion date corresponding to a provided probability of survival. See doi: 10.1111/biom.13472, Supporting Information, Section A.4.

**Usage**

```
seroconversion_inverse_survival_function(u, e, hazard_alpha, hazard_beta)
```

**Arguments**

- `u` a vector of seroconversion survival probabilities
- `e` a vector of time differences between study start and enrollment (in years)
- `hazard_alpha` the instantaneous hazard of seroconversion on the study start date
- `hazard_beta` the change in hazard per year after study start date

**Value**

numeric vector of time differences between study start and seroconversion (in years)
References


simulate_interval_censoring

Simulate a dataset with interval-censored seroconversion dates

Description

simulate_interval_censoring generates a simulated data set from a data-generating model based on the typical structure of a cohort study of HIV biomarker progression, as described in Morrison et al (2021); doi: 10.1111/biom.13472.

Usage

simulate_interval_censoring(
  study_cohort_size = 4500,
  hazard_alpha = 1,
  hazard_beta = 0.5,
  preconversion_interval_length = 84,
  theta = c(0.986, -3.88),
  probability_of_ever_seroconverting = 0.05,
  years_in_study = 10,
  max_scheduling_offset = 7,
  days_from_study_start_to_recruitment_end = 365,
  study_start_date = lubridate::ymd("2001-01-01")
)

Arguments

study_cohort_size
  the number of participants to simulate (N_0 in the paper)

hazard_alpha
  the hazard (instantaneous risk) of seroconversion at the start date of the cohort study for those participants at risk of seroconversion

hazard_beta
  the change in hazard per calendar year

preconversion_interval_length
  the number of days between tests for seroconversion

theta
  the parameters of a logistic model (with linear functional term) specifying the probability of MAA-positive biomarkers as a function of time since seroconversion

probability_of_ever_seroconverting
  the probability that each participant is at risk of HIV seroconversion

years_in_study
  the duration of follow-up for each participant
simulate_interval_censoring

max_scheduling_offset
the maximum divergence of pre-seroconversion followup visits from the pre-
scribed schedule

days_from_study_start_to_recruitment_end
the length of the recruitment period

study_start_date
the date when the study starts recruitment ("d_0" in the main text). The value
of this parameter does not affect the simulation results; it is only necessary as a
reference point for generating E, L, R, O, and S.

Value
A list containing the following two tibbles:

- pt_data: a tibble of participant-level information, with the following columns:
  - ID: participant ID
  - E: enrollment date
  - L: date of last HIV test prior to seroconversion
  - R: date of first HIV test after seroconversion
- obs_data: a tibble of longitudinal observations with the following columns:
  - ID: participant ID
  - O: dates of biomarker sample collection
  - Y: MAA classifications of biomarker samples

References

Examples
study_data <- simulate_interval_censoring()
participant_characteristics <- study_data$pt_data
longitudinal_observations <- study_data$obs_data
Index

build_phi_function_from_coefs, 2
compute_mu, 2
fit_joint_model, 3, 10
fit_midpoint_model, 5
fit_uniform_model, 6
plot_CDF, 7
plot_phi_curves, 8
rwicc, 10
seroconversion_inverse_survival_function, 10
simulate_interval_censoring, 10, 11