

Package ‘iPEB’

September 17, 2026

Type Package

Title Improved Parametric Empirical Bayes for Longitudinal Biomarker Analysis

Version 0.1.1

Description Extends parametric empirical Bayes (PEB) for longitudinal biomarker screening with a time-gap-aware standardization layer, covariate adjustment, and objective-driven multi-marker weighting. The layer models each subject's biomarker history with a random intercept (and an optional random slope) and autocorrelated, gap-scaled residuals, so that prediction uncertainty grows with the time between visits and per-visit specificity is preserved under irregular sampling. Marker weights are learned to optimize a user-selected clinical objective -- maximizing sensitivity at a fixed specificity, extending detection lead time, or a combined objective -- with optional feature selection and a choice of scalar or multivariate combiner. Functions for fitting, prediction, and evaluation (sensitivity, lead time, and specificity at chosen operating points) are provided. A manuscript describing the method is in preparation.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Depends R (>= 4.0.0)

Imports nlme, stats, graphics, utils

Suggests testthat (>= 3.0.0), knitr, rmarkdown, shiny

VignetteBuilder knitr

URL <https://github.com/bitansa/iPEB>

BugReports <https://github.com/bitansa/iPEB/issues>

Config/testthat/edition 3

Config/roxygen2/version 8.1.0

NeedsCompilation no

Author Bitan Sarkar [aut, cre, cph] (ORCID:
 <<https://orcid.org/0009-0000-5178-5934>>),
 Ana Maria Kenney [aut],
 James P. Long [aut],
 Johannes F. Fahrman [aut],
 Samir Hanash [aut],
 Kim-Anh Do [aut],
 Ehsan Irajizad [aut]
Maintainer Bitan Sarkar <bitansarkar010899@gmail.com>
Repository CRAN
Date/Publication 2026-09-17 13:20:08 UTC

Contents

evaluate	2
ipeb	3
ipeb_example	5
ipeb_innovations	6
ipeb_run	7
plot.ipeb	9
predict.ipeb	9
run_app	10
Index	12

evaluate	<i>Evaluate a fitted iPEB model</i>
----------	-------------------------------------

Description

Computes per-patient sensitivity and median lead time with per-visit specificity at one or more operating points. Operating thresholds are calibrated on the training control visits stored in the fitted object and applied unchanged to newdata, so every reported number is out-of-sample when newdata are held-out subjects.

Usage

```
evaluate(object, ...)  
  
## S3 method for class 'ipeb'  
evaluate(object, newdata, specificities = NULL, window = NULL, ...)
```

Arguments

<code>object</code>	A fitted "ipeb" object.
<code>...</code>	Passed to methods.
<code>newdata</code>	A data frame with the fitting schema, including the case indicator (needed to score sensitivity and lead time).
<code>specificities</code>	Numeric vector of target specificities. Defaults to the specificity the model was optimized at.
<code>window</code>	Optional evaluation detection window in months. If NULL (default), the window used at fitting is reused; supply a value (or Inf for the whole trajectory) to report at a different window than training.

Value

A data frame with one row per requested specificity and columns `specificity` (target), `sensitivity`, `lead_time` (median, years), `realized_specificity` (achieved on `newdata`), and `auc` (subject-level, specificity-independent).

Examples

```
data(ipeb_example)
train <- subset(ipeb_example, split == "train")
test  <- subset(ipeb_example, split == "test")

fit <- ipeb(train, markers = c("m1", "m2", "m3"),
            innovation = "iid", slope = "off")
evaluate(fit, test, specificities = c(0.90, 0.95, 0.99))
```

ipeb

*Fit an improved Parametric Empirical Bayes (iPEB) model***Description**

Fits the iPEB model on longitudinal biomarker data: a time-gap-aware standardization layer, optional covariate adjustment, objective-driven multi-marker weighting, optional feature selection, and an automatically chosen scalar or multivariate combiner. The returned object can score and evaluate new subjects with `predict.ipeb` and `evaluate`.

Usage

```
ipeb(
  data,
  markers,
  id = "id",
  case = "case",
```

```

time = "time",
time_to_dx = "time_to_dx",
covariates = character(0),
objective = c("sensitivity", "leadtime", "combined"),
alpha = 0.95,
window = Inf,
slope = c("auto", "on", "off"),
innovation = c("auto", "ar1", "iid"),
select = c("none", "backward"),
n_markers = NULL,
validation_frac = 0.25,
seed = NULL
)

```

Arguments

<code>data</code>	A long-format data frame with one row per subject-visit.
<code>markers</code>	Character vector of biomarker column names.
<code>id, case, time, time_to_dx</code>	Column names for the subject identifier, the case indicator (1 = case, 0 = control), the visit time, and each visit's time before diagnosis (in days; used to measure lead time).
<code>covariates</code>	Character vector of covariate column names to adjust for (default none).
<code>objective</code>	Clinical objective: "sensitivity" (default), "leadtime", or "combined". Each fixes an internal penalty profile rather than requiring manual tuning.
<code>alpha</code>	Operating specificity at which the combiner is optimized (default 0.95).
<code>window</code>	Optional detection window in months restricting the objective to a fixed pre-diagnostic horizon. The default, Inf, uses the whole pre-diagnostic trajectory (no window).
<code>slope</code>	Random-slope layer: "auto" (default; on when subjects have enough visits), "on", or "off".
<code>innovation</code>	Residual model: "auto" (default) or "ar1" use gap-scaled AR(1)/OU innovations; "iid" uses independent innovations.
<code>select</code>	Feature selection by objective-driven backward elimination: "none" (default) or "backward".
<code>n_markers</code>	Optional target panel size when select is not "none".
<code>validation_frac</code>	Fraction of training subjects held out as an internal validation slice to choose the combiner variant and drive selection (default 0.25).
<code>seed</code>	Optional single number. 'ipeb()' draws an internal validation split of the training subjects (and, when 'select' is not "none", uses it for feature selection), so repeated calls on identical data can return slightly different weights. Supplying 'seed' fixes that draw and makes a fit exactly reproducible; the default 'NULL' leaves the RNG untouched.

Value

An object of class "ipeb": a list with the fitted weights, selected markers, chosen variant, layer parameters, and the training control scores used to calibrate operating thresholds.

See Also

[predict.ipeb](#), [evaluate](#), [ipeb_run](#)

Examples

```
data(ipeb_example)
train <- subset(ipeb_example, split == "train")

fit <- ipeb(train, markers = c("m1", "m2", "m3"),
            id = "id", case = "case", time = "time",
            time_to_dx = "time_to_dx", objective = "sensitivity",
            innovation = "iid", slope = "off")

fit
```

ipeb_example

Synthetic longitudinal biomarker example

Description

A small, entirely simulated longitudinal cohort used to illustrate the package. Each subject contributes several irregularly spaced visits with three biomarkers. In cases, marker m1 rises from roughly two years before diagnosis and m2 rises late (a few months before), while m3 is uninformative; controls have no pre-diagnostic rise. No real patient data are used.

Usage

```
ipeb_example
```

Format

A data frame with one row per subject-visit and columns:

id Subject identifier.
case Case indicator (1 = case, 0 = control).
time Visit time in years since the subject's first visit.
time_to_dx Days from the visit to diagnosis (positive).
m1, m2, m3 Biomarker measurements.
split Suggested "train"/"test" split by subject.

Source

Simulated by data-row/make_example.R.

ipeb_innovations	<i>Time-gap-aware standardized innovations</i>
------------------	--

Description

Returns the innovation-standardized residuals from the iPEB layer: for each marker, a random intercept (and optional slope) model with autocorrelated, gap-scaled residuals is estimated on the healthy reference subjects, and every visit is scored by sequential one-step prediction (history to innovation, divided by its prediction standard deviation). The prediction variance grows with the gap since the previous visit, so the innovations are comparable across irregular schedules.

Usage

```
ipeb_innovations(
  data,
  markers,
  reference_ids = NULL,
  id = "id",
  case = "case",
  time = "time",
  covariates = character(0),
  slope = c("off", "on"),
  innovation = c("iid", "ar1")
)
```

Arguments

data	A long-format data frame, one row per subject-visit.
markers	Character vector of biomarker column names to standardize.
reference_ids	Optional vector of subject ids whose healthy (control) visits define the reference for estimating the layer. Defaults to all subjects in data.
id, case, time	Column names for the subject id, the case indicator (1 = case, 0 = control), and the visit time.
covariates	Character vector of covariate columns to adjust for.
slope	Random slope: "off" (default) or "on".
innovation	Residual model: "iid" (default) or "ar1" for gap-scaled AR(1)/OU residuals.

Details

This is the layer that underlies [ipeb](#). It is exposed so that history-adjusted baselines (for example, a fixed-panel PEB score, a single-marker PEB score, or a principal component of the innovations) can be built directly from the same layer.

Value

A numeric matrix with one row per row of data and one column per marker (named by marker), containing the standardized innovations.

See Also[ipeb](#)**Examples**

```
data(ipeb_example)
train <- subset(ipeb_example, split == "train")

R <- ipeb_innovations(train, markers = c("m1", "m2", "m3"))
head(R)
```

ipeb_run*Fit and evaluate iPEB in one call*

Description

Convenience wrapper that fits [ipeb](#) on train and evaluates the fitted model on test.

Usage

```
ipeb_run(
  train,
  test,
  markers,
  id = "id",
  case = "case",
  time = "time",
  time_to_dx = "time_to_dx",
  covariates = character(0),
  objective = c("sensitivity", "leadtime", "combined"),
  alpha = 0.95,
  window = Inf,
  slope = c("auto", "on", "off"),
  innovation = c("auto", "ar1", "iid"),
  select = c("none", "backward"),
  n_markers = NULL,
  validation_frac = 0.25,
  specificities = NULL,
  eval_window = NULL,
  seed = NULL
)
```

Arguments

<code>train</code>	Training data frame.
<code>test</code>	Test data frame (same schema as <code>train</code>).
<code>markers</code>	Character vector of biomarker column names.
<code>id, case, time, time_to_dx</code>	Column names for the subject identifier, the case indicator (1 = case, 0 = control), the visit time, and each visit's time before diagnosis (in days; used to measure lead time).
<code>covariates</code>	Character vector of covariate column names to adjust for (default none).
<code>objective</code>	Clinical objective: "sensitivity" (default), "leadtime", or "combined". Each fixes an internal penalty profile rather than requiring manual tuning.
<code>alpha</code>	Operating specificity at which the combiner is optimized (default 0.95).
<code>window</code>	Optional detection window in months restricting the objective to a fixed pre-diagnostic horizon. The default, Inf, uses the whole pre-diagnostic trajectory (no window).
<code>slope</code>	Random-slope layer: "auto" (default; on when subjects have enough visits), "on", or "off".
<code>innovation</code>	Residual model: "auto" (default) or "ar1" use gap-scaled AR(1)/OU innovations; "iid" uses independent innovations.
<code>select</code>	Feature selection by objective-driven backward elimination: "none" (default) or "backward".
<code>n_markers</code>	Optional target panel size when <code>select</code> is not "none".
<code>validation_frac</code>	Fraction of training subjects held out as an internal validation slice to choose the combiner variant and drive selection (default 0.25).
<code>specificities</code>	Numeric vector of target specificities for evaluation. Defaults to <code>alpha</code> .
<code>eval_window</code>	Optional evaluation detection window in months (passed to evaluate); NULL reuses the training window.
<code>seed</code>	Optional single number passed to ipeb to fix its internal validation split, making the fit exactly reproducible.

Value

A list with two elements: `fit` (the fitted "ipeb" object) and `evaluation` (the data frame returned by [evaluate](#) on test).

See Also

[ipeb](#), [evaluate](#)

Examples

```
data(ipeb_example)
train <- subset(ipeb_example, split == "train")
test  <- subset(ipeb_example, split == "test")

res <- ipeb_run(train, test, markers = c("m1", "m2", "m3"),
               objective = "sensitivity", innovation = "iid", slope = "off",
               specificities = c(0.90, 0.95, 0.99))
res$evaluation
```

plot.ipeb	<i>Plot the fitted iPEB marker weights</i>
-----------	--

Description

Plot the fitted iPEB marker weights

Usage

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

Arguments

x	A fitted "ipeb" object.
...	Passed to barplot .

Value

The fitted object, invisibly.

predict.ipeb	<i>Score new subjects with a fitted iPEB model</i>
--------------	--

Description

Computes the iPEB composite score for each visit in newdata using the layer parameters and combiner weights stored in a fitted model. A subject is scored from its own trajectory against the training healthy reference; higher scores indicate stronger evidence of disease.

Usage

```
## S3 method for class 'ipeb'
predict(object, newdata, ...)
```

Arguments

object	A fitted "ipeb" object from ipeb .
newdata	A data frame with the same columns used at fitting (the selected markers, co-variates, id, time, and time-to-diagnosis; the case column is optional here).
...	Unused.

Value

A numeric vector of iPEB scores, one per row of newdata.

See Also

[ipeb](#), [evaluate](#)

Examples

```
data(ipeb_example)
train <- subset(ipeb_example, split == "train")
test  <- subset(ipeb_example, split == "test")

fit <- ipeb(train, markers = c("m1", "m2", "m3"),
            innovation = "iid", slope = "off")
scores <- predict(fit, test)
```

run_app

Launch the iPEB Shiny explorer

Description

Opens the companion Shiny application bundled with the package. The app lets you load the built-in synthetic example (or upload a long-format CSV), choose the objective, operating specificity, windowing, layer options and feature selection, then fit [ipeb](#) and view the marker weights and the held-out evaluation interactively.

Usage

```
run_app(...)
```

Arguments

... Additional arguments passed to [runApp](#) (for example `launch.browser` or `port`).

Details

The app requires the **shiny** package (a suggested dependency); install it with `install.packages("shiny")` if needed. A hosted, browser-based version for users without R will be linked from the package README once deployed.

Value

Invisibly NULL; called for the side effect of launching the app.

See Also

[ipeb](#)

Examples

```
if (interactive()) {  
  run_app()  
}
```

Index

* **datasets**

ipeb_example, [5](#)

barplot, [9](#)

evaluate, [2](#), [3](#), [5](#), [8](#), [10](#)

ipeb, [3](#), [6–8](#), [10](#), [11](#)

ipeb_example, [5](#)

ipeb_innovations, [6](#)

ipeb_run, [5](#), [7](#)

plot.ipeb, [9](#)

predict.ipeb, [3](#), [5](#), [9](#)

run_app, [10](#)

runApp, [10](#)