Package: pvEBayes (via r-universe)

May 20, 2026

Title Empirical Bayes Methods for Pharmacovigilance

Version 0.3.0

Maintainer Yihao Tan <yihaotan@buffalo.edu>

Description A suite of empirical Bayes methods to use in pharmacovigilance. Contains various model fitting and post-processing functions. For more details see Tan et al. (2025) <doi:10.1002/sim.70195>, <doi:10.48550/arXiv.2512.01057>; Koenker and Mizera (2014) <doi:10.1080/01621459.2013.869224>; Efron (2016) <doi:10.1093/biomet/asv068>.

License GPL-3

Encoding UTF-8

Roxygen list(markdown = TRUE, roclets = c (` `namespace", ` `rd", ` `srr::srr_stats_roclet"))

RoxygenNote 7.3.2

Depends R (>= 3.6.0)

Imports data.table, ggdist, ggfittext, ggplot2, glue, graphics, magrittr, methods, stats, wacolors, Rcpp, splines, CVXR, cli

LinkingTo Rcpp, RcppEigen

LazyData true

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

Config/testthat/edition 3

URL https://github.com/YihaoTancn/pvEBayes, https://yihaotancn.github.io/pvEBayes/

BugReports https://github.com/YihaoTancn/pvEBayes/issues

VignetteBuilder knitr

Config/pak/sysreqs cmake libgmp3-dev make libicu-dev libjpeg-dev libpng-dev libxml2-dev libssl-dev pkg-config libclang-dev

Repository https://yihaotancn.r-universe.dev

Date/Publication 2026-05-20 18:07:25 UTC

RemoteUrl https://github.com/yihaotancn/pvebayes

RemoteRef HEAD

RemoteSha 1bbbdfe60c4349b15859ef73f9b0b71dc49aff69

Contents

`pvEBayes-package` *A suite of empirical Bayes methods to use in pharmacovigilance.*

Description

`pvEBayes` provides a collection of parametric and non-parametric empirical Bayes methods implementation for pharmacovigilance (including signal detection and signal estimation) on spontaneous reporting systems (SRS) data.

An SRS dataset catalogs AE reports on I AE rows across J drug columns. Let N_{ij} denote the number of reported cases for the i-th AE and the j-th drug, where $i = 1, ..., I$ and $j = 1, ..., J$. We assume that for each AE-drug pair, $N_{ij} \sim \text{Poisson}(\lambda_{ij} E_{ij})$, where E_{ij} is expected baseline value measuring the expected count of the AE-drug pair when there is no association between i-th AE

and *j*-th drug. The parameter $\lambda_{ij} \geq 0$ represents the relative reporting ratio, the signal strength, for the (i, j)-th pair measuring the ratio of the actual expected count arising due to dependence to the null baseline expected count. Current disproportionality analysis mainly focuses on *signal detection* which seeks to determine whether the observation N_{ij} is substantially greater than the corresponding null baseline E_{ij}. Under the Poisson model, that is to say, its signal strength λ_{ij} is significantly greater than 1.

In addition to *signal detection*, Tan et al. (*Stat. in Med.*, 2025) broaden the role of disproportionality to *signal estimation*. The use of the flexible non-parametric empirical Bayes models enables more nuanced empirical Bayes posterior inference (parameter estimation and uncertainty quantification) on signal strength parameter $\{\lambda_{ij}\}$. This allows researchers to distinguish AE-drug pairs that would appear similar under a binary signal detection framework. For example, the AE-drug pairs with signal strengths of 1.5 and 4.0 could both be significantly greater than 1 and detected as a signal. Such differences in signal strength may have distinct implications in medical and clinical contexts.

The methods included in `pvEBayes` differ by their assumptions on the prior distribution. Implemented methods include the Gamma-Poisson Shrinker (GPS), Koenker-Mizera (KM) method, Efron's nonparametric empirical Bayes approach, the K-gamma model, and the general-gamma model.

The GPS model uses two gamma mixture prior by assuming the signal/non-signal structure in SRS data. However, in real-world setting, signal strengths (λ_{ij}) are often heterogeneous and thus follows a multi-modal distribution, making it difficult to assume a parametric prior. Non-parametric empirical Bayes models (KM, Efron, K-gamma and general-gamma) address this challenge by utilizing a flexible prior with general mixture form and estimating the prior distribution in a data-driven way.

`pvEBayes` offers the first implemention of the bi-level Expectation Conditional Maximization (ECM) algorithm proposed by Tan et al. (2025) for efficient parameter estimation in gamma mixture prior based models: GPS K-gamma and general-gamma.

The KM method has an existing implementation in the `REBayes` package, but it relies on Mosek, a commercial convex optimization solver, which may limit accessibility due to licensing issue. `pvEBayes` provides a alternative fully open-source implementation of the KM method using `CVXR`.

Efron's method also has a general nonparametric empirical Bayes implementation in the `deconvolveR` package; however, that implementation does not support an exposure or offset parameter in the Poisson model, which corresponds to the expected null value E_{ij}. In `pvEBayes`, the implementation of the Efron's method is adapted and modified from `deconvolveR` to support E_{ij} in Poisson model.

For a detailed introduction to `pvEBayes`, see Tan et al. (*arxiv*, 2025) and package Vignette.

Author(s)

Yihao Tan, Marianthi Markatou, Saptarshi Chakraborty and Raktim Mukhopadhyay.

Maintainer: Yihao Tan <yihaotan@buffalo.edu>

References

Tan Y, Markatou M and Chakraborty S. Flexible Empirical Bayesian Approaches to Pharmacovigilance for Simultaneous Signal Detection and Signal Strength Estimation in Spontaneous Reporting Systems Data. *Statistics in Medicine.* 2025; 44: 18-19, https://doi.org/10.1002/sim.70195.

Tan Y, Markatou M and Chakraborty S. pvEBayes: An R Package for Empirical Bayes Methods in Pharmacovigilance. *arXiv*:2512.01057 (stat.AP). https://doi.org/10.48550/arXiv.2512.01057

Koenker R, Mizera I. Convex Optimization, Shape Constraints, Compound Decisions, and Empirical Bayes Rules. *Journal of the American Statistical Association* 2014; 109(506): 674–685, https://doi.org/10.1080/01621459.2013.869224

Efron B. Empirical Bayes Deconvolution Estimates. *Biometrika* 2016; 103(1); 1-20, https://doi.org/10.1093/biomet/asv068

DuMouchel W. Bayesian data mining in large frequency tables, with an application to the FDA spontaneous reporting system. *The American Statistician.* 1999; 1;53(3):177-90.

See Also

Useful links:

- https://github.com/YihaoTancn/pvEBayes
- https://yihaotancn.github.io/pvEBayes/
- Report bugs at https://github.com/YihaoTancn/pvEBayes/issues

---

`AIC.pvEBayes` *Obtain Akaike Information Criterion (AIC) for a pvEBayes object*

Description

This function defines the S3 `AIC` method for objects of class `pvEBayes`. It extracts the Akaike Information Criterion (AIC) from a fitted model.

Usage

```
## S3 method for class 'pvEBayes'
AIC(object, ..., k = 2)
```

Arguments

`object`	a `pvEBayes` object, which is the output of the function pvEBayes or pvEBayes_tune.
`...`	other input parameters. Currently unused.
`k`	numeric, the penalty per parameter to be used; the default k = 2 is the classical AIC.

Value

numeric, AIC score for the resulting model.

Examples

```
fit <- pvEBayes(
  contin_table = statin2025_44, model = "general-gamma",
  alpha = 0.3, n_posterior_draws = NULL
)

AIC_score <- AIC(fit)
```

BIC.pvEBayes *Obtain Bayesian Information Criterion (BIC) for a pvEBayes object*

Description

This function defines the S3 BIC method for objects of class pvEBayes. It extracts the Bayesian Information Criterion (BIC) from a fitted model.

Usage

```
## S3 method for class 'pvEBayes'
BIC(object, ...)
```

Arguments

object	a pvEBayes object, which is the output of the function pvEBayes or pvEBayes_tune.
...	other input parameters. Currently unused.

Value

numeric, BIC score for the resulting model.

Examples

```
fit <- pvEBayes(
  contin_table = statin2025_44, model = "general-gamma",
  alpha = 0.3, n_posterior_draws = NULL
)

BIC_score <- BIC(fit)
```

estimate_null_expected_count *Estimate expected null baseline count based on reference row and column*

Description

This function estimates the expected null baseline count (E_{ij}) for each AE-drug combination under the assumption of independence between rows and columns. The expected count is calculated using a reference row (other AEs) and reference column (other drugs). This null baseline is typically used in empirical Bayes modeling of **pvEBayes** package for signal detection and estimation in spontaneous reporting system (SRS) data.

Usage

```
estimate_null_expected_count(contin_table)
```

Arguments

contin_table	an IxJ contingency table showing pairwise counts of adverse events for I AEs (along the rows) and J drugs (along the columns). The reference row "Other AEs" and the reference column "Other drugs" need to be the I-th row and J-th column respectively.

Details

This null value estimator is proposed by Tan et al. (2025).

Value

an `nrow(contin_table)` by `ncol(contin_table)` matrix.

References

Tan Y, Markatou M and Chakraborty S. Flexible Empirical Bayesian Approaches to Pharmacovigilance for Simultaneous Signal Detection and Signal Strength Estimation in Spontaneous Reporting Systems Data. *Statistics in Medicine.* 2025; 44: 18-19, https://doi.org/10.1002/sim.70195.

Examples

```
estimate_null_expected_count(statin2025_44)
```

---

extract_all_fitted_models

Extract all fitted models from a tuned pvEBayes Object

Description

This function retrieves the list of all fitted models from a pvEBayes_tuned object, which is the output of the `pvEBayes_tune()` function.

Usage

```
extract_all_fitted_models(object)
```

Arguments

object	An object of class `pvEBayes_tuned`, usually returned by `pvEBayes_tune`. This function will throw an error if the input is not of the correct class.

Value

A list containing the results of each model fitted during the tuning process.

Examples

```
valid_matrix <- matrix(c(1, 2, 3, 4, 5, 6, 7, 8), nrow = 2)
rownames(valid_matrix) <- c("AE_1", "AE_2")
colnames(valid_matrix) <- c("drug_1", "drug_2", "drug_3", "drug_4")

tuned_object <- pvEBayes_tune(valid_matrix,
  model = "general-gamma",
  return_all_fit = TRUE
)
extract_all_fitted_models(tuned_object)
```

---

eyeplot_pvEBayes *Generate an eyeplot showing the distribution of posterior draws for selected drugs and adverse events*

Description

This function creates an eyeplot to visualize the posterior distributions of λ_{ij} for selected AEs and drugs. The plot displays posterior median, 90 percent credible interval for each selected AE-drug combination.

Usage

```
eyeplot_pvEBayes(
  x,
  num_top_AEs = 10,
  num_top_drugs = 8,
  specified_AEs = NULL,
  specified_drugs = NULL,
  N_threshold = 1,
  text_shift = 4,
  x_lim_scalar = 1.3,
  text_size = 3,
  log_scale = FALSE
)
```

Arguments

x	a pvEBayes object, which is the output of the function pvEBayes or pvEBayes_tune.
num_top_AEs	a number of most significant AEs appearing in the plot. Default to 10.
num_top_drugs	a number of most significant drugs appearing in the plot. Default to 7.

specified_AEs	a vector of AE names that are specified to appear in the plot. If a vector of AEs is given, argument num_top_AEs will be ignored.
specified_drugs	a vector of drug names that are specified to appear in the plot. If a vector of drugs is given, argument num_top_drugs will be ignored.
N_threshold	a integer greater than 0. Any AE-drug combination with observation smaller than N_threshold will be filtered out.
text_shift	numeric. Controls the relative position of text labels, (e.g., "N = 1", "E = 2"). A larger value shifts the "E = 2" further away from its original position.
x_lim_scalar	numeric. An x-axis range scalar that ensures text labels are appropriately included in the plot.
text_size	numeric. Controls the size of text labels, (e.g., "N = 1", "E = 2").
log_scale	logical. If TRUE, the eye plot displays the posterior distribution of $\log(\lambda_{ij})$ for the selected AEs and drugs.

Value

a ggplot2 object.

Examples

```
fit <- pvEBayes(
  contin_table = statin2025_44, model = "general-gamma",
  alpha = 0.3, n_posterior_draws = 1000
)

AE_names <- rownames(statin2025_44)[1:6]
drug_names <- colnames(statin2025_44)[-7]

eyeplot_pvEBayes(
  x = fit
)
```

---

faers_opioid_mental *FDA opioid dataset with 243 mental-related adverse events*

Description

An adverse event-drug count dataset (contingency table) obtained from the FDA FAERS database for the quarters 2014Q3 - 2020Q4.

Usage

```
faers_opioid_mental
```

Format

An object of class `matrix` (inherits from `array`) with 244 rows and 6 columns.

Details

Data are stored in the form of a contingency table, with drugs listed across the columns and the 243 AEs presented across the rows. Each cell in the contingency table represents the total report counts associated with that (AE, drug) pair and detected in the FDA FAERS database during 2014Q3 - 2020Q4.

The dataset catalogs 5 opioid drugs (across columns):

Codeine, Fentanyl, Oxycodone, Pentazocine and Tramadol.

Source

https://fis.fda.gov/extensions/FPD-QDE-FAERS/FPD-QDE-FAERS.html

References

Tan Y, Markatou M and Chakraborty S. A Review of Statistical Methods for Spontaneous Reporting System Data Mining: Signal Detection and Beyond. *arXiv*:2604.18898 (stat.AP). https://doi.org/10.48550/arXiv.2604.18898

---

gbca2025 *FDA GBCA dataset with 1328 adverse events*

Description

An adverse event-drug count dataset (contingency table) obtained from the FDA FAERS database for the quarters 2021Q1 - 2024Q4.

Usage

```
gbca2025
```

Format

An object of class `matrix` (inherits from `array`) with 1328 rows and 8 columns.

Details

Data are stored in the form of a contingency table, with drugs listed across the columns and the 1328 AEs presented across the rows. Each cell in the contingency table represents the total report counts associated with that (AE, drug) pair and detected in the FDA FAERS database during 2021Q1 - 2024Q4.

The dataset catalogs 7 Gadolinium-Based Contrast Agents (GBCAs):

Gadobenate, Gadobutrol, Gadodiamide, Gadopentetate, Gadoterate, Gadoteridol, Gadoxetate

The 1328 adverse events presented across the rows are AEs that contain at least one report for the 7 GBCA drugs during 2021Q1 - 2024Q4.

This dataset is an updated version of gbca from the pvLRT package which collects the same scope of AEs for 7 gbca drugs for quarters 2014Q3 - 2020Q4.

Source

https://fis.fda.gov/extensions/FPD-QDE-FAERS/FPD-QDE-FAERS.html

---

gbca2025_69 *FDA GBCA dataset with 69 adverse events*

Description

An adverse event-drug count dataset (contingency table) obtained from the FDA FAERS database for the quarters 2021Q1 - 2024Q4

Usage

```
gbca2025_69
```

Format

An object of class `matrix` (inherits from `array`) with 70 rows and 8 columns.

Details

Data are stored in the form of a contingency table, with drugs listed across the columns and the 69 AEs presented across the rows. Each cell in the contingency table represents the total report counts associated with that (AE, drug) pair and detected in the FDA FAERS database during 2021Q1 - 2024Q4.

The dataset catalogs 7 Gadolinium-Based Contrast Agents (GBCAs) (across columns):

Gadobenate, Gadobutrol, Gadodiamide, Gadopentetate, Gadoterate, Gadoteridol, Gadoxetate.

The 69 adverse events presented across the rows are selected from 1328 AEs of gbca2025 which are related to the brain or neural system. Other AEs are collapsed to one reference row: "Other AEs".

Source

https://fis.fda.gov/extensions/FPD-QDE-FAERS/FPD-QDE-FAERS.html

`generate_contin_table` *Generate random contingency tables based on a reference table embedded signals,and possibly with zero inflation*

Description

This function generates random contingency tables that resemble a given reference table, with the option to embed signals and zero-inflation.

Usage

```
generate_contin_table(
  n_table = 1,
  ref_table,
  signal_mat = NULL,
  Variation = FALSE,
  zi_indic_mat = NULL
)
```

Arguments

`n_table`	a number of random matrices to generate.
`ref_table`	a reference table used as the basis for generating random tables.
`signal_mat`	a numeric matrix of the same dimension as the reference table (ref_table). The entry at position (i, j) in signal_mat represents the signal strength between the i-th adverse event and the j-th drug. By default, each pair is assigned a value of 1, indicating no signal for that pair.
`Variation`	logical. Include random noises to sig_mat while generating random tables. Default to FALSE. If set to TRUE, n_table of sig_mat incorporating the added noise will also be returned.
`zi_indic_mat`	logical matrix of the same size as ref_table indicating the positions of structural zero.

Value

A list of length `n_table`, with each entry being a `nrow(ref_table)` by `ncol(ref_table)` matrix.

References

Tan Y, Markatou M and Chakraborty S. Flexible Empirical Bayesian Approaches to Pharmacovigilance for Simultaneous Signal Detection and Signal Strength Estimation in Spontaneous Reporting Systems Data. *Statistics in Medicine.* 2025; 44: 18-19, https://doi.org/10.1002/sim.70195.

Examples

```
set.seed(1)
ref_table <- statin2025_44


# set up signal matrix with one signal at entry (1,1)
sig_mat <- matrix(1, nrow(ref_table), ncol(ref_table))
sig_mat[1, 1] <- 2

# set up structural zero matrix
Z <- matrix(0, nrow(ref_table), ncol(ref_table))
Z[5, 1] <- 1

simu_table <- generate_contin_table(ref_table,
  signal_mat = sig_mat,
  n_table = 1,
  Variation = TRUE,
  zi_indic_mat = Z
)[[1]][[1]]
```

---

get_posterior_prob *Obtain posterior probability of being a signal*

Description

Obtain posterior probability of being a signal

Usage

```
get_posterior_prob(obj, cutoff_signal = 1.001)
```

Arguments

obj	a pvEBayes object, which is the output of the function pvEBayes or pvEBayes_tune.
cutoff_signal	numeric. Threshold for signal detection. An AE-drug combination is classified as a detected signal if its 5th posterior percentile exceeds this threshold.

Value

a matrix

heatmap_pvEBayes *Generate a heatmap plot visualizing posterior probabilities for selected drugs and adverse events*

Description

This function generates a heatmap to visualize the posterior probabilities of being a signal for selected AEs and drugs.

Usage

```
heatmap_pvEBayes(
  x,
  num_top_AEs = 10,
  num_top_drugs = 8,
  specified_AEs = NULL,
  specified_drugs = NULL,
  cutoff_signal = NULL
)
```

Arguments

x	a pvEBayes object, which is the output of the function pvEBayes or pvEBayes_tune.
num_top_AEs	number of most significant AEs appearing in the plot. Default to 10.
num_top_drugs	number of most significant drugs appearing in the plot. Default to 7.
specified_AEs	a vector of AE names that are specified to appear in the plot. If a vector of AEs is given, argument num_top_AEs will be ignored.
specified_drugs	a vector of drug names that are specified to appear in the plot. If a vector of drugs is given, argument num_top_drugs will be ignored.
cutoff_signal	numeric. Threshold for signal detection. An AE-drug combination is classified as a detected signal if its 5th posterior percentile exceeds this threshold.

Value

a ggplot2 object.

Examples

```
fit <- pvEBayes(
  contin_table = statin2025_44, model = "general-gamma",
  alpha = 0.3, n_posterior_draws = 1000
)


heatmap_pvEBayes(
```

```
  x = fit,
  num_top_AEs = 10,
  num_top_drugs = 8,
  specified_AEs = NULL,
  specified_drugs = NULL,
  cutoff_signal = 1.001
)
```

`logLik.pvEBayes` *Extract log marginal likelihood for a pvEBayes object*

Description

This function defines the S3 `logLik` method for objects of class `pvEBayes`. It extracts the log marginal likelihood from a fitted model.

Usage

```
## S3 method for class 'pvEBayes'
logLik(object, ...)
```

Arguments

`object`	a `pvEBayes` object, which is the output of the function pvEBayes or pvEBayes_tune.
`...`	other input parameters. Currently unused.

Value

returns log marginal likelihood of a pvEBayes object.

Examples

```
fit <- pvEBayes(
  contin_table = statin2025_44, model = "general-gamma",
  alpha = 0.3, n_posterior_draws = NULL
)

logLik(fit)
```

plot.pvEBayes *Plotting method for a pvEBayes object*

Description

This function defines the S3 plot method for objects of class pvEBayes.

Usage

```
## S3 method for class 'pvEBayes'
plot(x, type = "eyeplot", ...)
```

Arguments

x	a pvEBayes object, which is the output of the function pvEBayes or pvEBayes_tune.
type	character string determining the type of plot to show. Available choices are "eyeplot" which calls eyeplot_pvEBayes and "heatmap" which calls heatmap_pvEBayes.
...	additional arguments passed to heatmap_pvEBayes or eyeplot_pvEBayes.

Value

A ggplot object.

Examples

```
obj <- pvEBayes(statin2025_44, model = "general-gamma", alpha = 0.5)
plot(obj, type = "eyeplot")
```

posterior_draws *Generate posterior draws for each AE-drug combination*

Description

This function generates posterior draws from the posterior distribution of λ_{ij} for each AE-drug combination, based on a fitted empirical Bayes model. The posterior draws can be used to compute credible intervals, visualize posterior distributions, or support downstream inference.

Usage

```
posterior_draws(obj, n_posterior_draws = 1000, verbose = TRUE)
```

Arguments

obj	a pvEBayes object, which is the output of the function pvEBayes or pvEBayes_tune.
n_posterior_draws	number of posterior draws for each AE-drug combination.
verbose	logical. If is TRUE (default), a progress bar is displayed to the console.

Value

The function returns an S3 object of class pvEBayes with posterior draws.

Examples

```
fit <- pvEBayes(
  contin_table = statin2025_44, model = "general-gamma",
  alpha = 0.3, n_posterior_draws = NULL
)

fit_with_draws <- posterior_draws(fit, n_posterior_draws = 1000)
```

---

print.pvEBayes *Print method for a pvEBayes object*

Description

This function defines the S3 print method for objects of class pvEBayes. It displays a concise description of the fitted model.

Usage

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

Arguments

x	a pvEBayes object, which is the output of the function pvEBayes or pvEBayes_tune.
...	other input parameters. Currently unused.

Value

Invisibly returns the input pvEBayes object.

Examples

```
obj <- pvEBayes(
  contin_table = statin2025_44, model = "general-gamma",
  alpha = 0.5, n_posterior_draws = 10000
)

print(obj)
```

---

pvEBayes *Fit a general-gamma, GPS, K-gamma, KM or efron model for a contingency table.*

Description

This function fits a non-parametric empirical Bayes model to an AE-drug contingency table using one of several empirical Bayes approaches with specified hyperparameter, if is required. Supported models include the "general-gamma", "GPS", "K-gamma", "KM", and "efron".

Usage

```
pvEBayes(
  contin_table,
  model = c("general-gamma", "K-gamma", "GPS", "KM", "efron"),
  alpha = NULL,
  K = NULL,
  p = NULL,
  c0 = NULL,
  maxi = NULL,
  tol_ecm = 1e-04,
  rtol_efron = 1e-10,
  rtol_KM = 1e-06,
  km_optimizer = c("ECOS", "CLARABEL", "SCS"),
  n_posterior_draws = 1000,
  E = NULL,
  message = TRUE,
  ...
)
```

Arguments

contin_table	an IxJ contingency table showing pairwise counts of adverse events for I AEs (along the rows) and J drugs (along the columns).
model	the model to fit. Available models are "general-gamma", "K-gamma", "GPS", "KM" and "efron". Default to "general-gamma". Note that the input for model is case-sensitive.

alpha	numeric between 0 and 1. The hyperparameter of "general-gamma" model. It is needed if "general-gamma" model is used. Small 'alpha' encourages shrinkage on mixture weights of the estimated prior distribution. See Tan et al. (2025) for further details.
K	a integer greater than or equal to 2 indicating the number of mixture components in the prior distribution. It is needed if "K-gamma" model is used. When K is unknown, please consider its extension "general-gamma" instead. See Tan et al. (2025) for further details.
p	a integer greater than or equal to 2. It is needed if "efron" mode is used. Larger p leads to smoother estimated prior distribution. See Narasimhan and Efron (2020) for detail.
c0	numeric and greater than 0. It is needed if "efron" mode is used. Large c0 encourage estimated prior distribution shrink toward discrete uniform. See Narasimhan and Efron (2020) for detail.
maxi	a upper limit of iteration for the ECM algorithm.
tol_ecm	a tolerance parameter used for the ECM stopping rule, defined as the absolute change in the joint marginal likelihood between two consecutive iterations. It is used when 'GPS', 'K-gamma' or 'general-gamma' model is fitted. Default to be 1e-4.
rtol_efron	a tolerance parameter used when 'efron' model is fitted. Default to 1e-10. See 'stats::nlminb' for detail.
rtol_KM	a tolerance parameter used when 'KM' model is fitted. Default to be 1e-6. See 'CVXR::solve' for detail.
km_optimizer	a character vector specifying the optimizer(s) in CVXR used to fit the KM model. Supported values are "ECOS", "CLARABEL", and "SCS". Note that the input for km_optimizer is case-sensitive. If multiple optimizers are supplied, they are tried sequentially and the first successfully fitted result is returned. Defaults to c("ECOS", "CLARABEL", "SCS"). See CVXR::psolve‘ for detail.
n_posterior_draws	a number of posterior draws for each AE-drug combination.
E	A matrix of expected counts under the null model for the SRS frequency table. If NULL (default), the expected counts are estimated from the SRS data using 'estimate_null_expected_count()'.
message	logical, indicating whether to print fitting information. Default to be TRUE.
...	additional parameters to be passed to optimizer for 'KM' model. See 'CVXR::solve' for detail.

Details

This function implements the ECM algorithm proposed by Tan et al. (2025), providing a stable and efficient implementation of Gamma-Poisson Shrinker(GPS), K-gamma and "general-gamma" methods for signal estimation and signal detection in Spontaneous Reporting System (SRS) data table.

Method "GPS" is proposed by DuMouchel (1999) and it is a parametric empirical Bayes model with a two gamma mixture prior distribution.

Methods "K-gamma" and "general-gamma" are non-parametric empirical Bayes models, introduced by Tan et al. (2025). The number of mixture components "K" needs to be prespecified when fitting a "K-gamma" model. When the number of mixture components is unknown, we recommend using the "general-gamma" model instead. For "general-gamma", the mixture weights are regularized by a Dirichlet hyper prior with hyperparameter $0 \leq \alpha < 1$ that controls the shrinkage strength. As "alpha" approaches 0, less non-empty mixture components exist in the fitted model. When $\alpha = 0$, the Dirichlet distribution is an improper prior still offering a reasonable posterior inference that represents the strongest shrinkage of the "general-gamma" model.

Parameter estimation for the "KM" model is formulated as a convex optimization problem. The objective function and constraints are modified from the **REBayes** package (see 'REBayes::KWDual()'). Parameter estimation is performed using the open-source convex optimization package **CVXR**.

The implementation of the "efron" model in this package is adapted from the **deconvolveR** package, developed by Bradley Efron and Balasubramanian Narasimhan. The original implementation in **deconvolveR** does not support an exposure or offset parameter in the Poisson model, which corresponds to the expected null value (E_{ij}) for each AE-drug combination. To address this, we modified the relevant code to allow for the inclusion of E_{ij} in the Poisson likelihood. In addition, we implemented a method for estimating the degrees of freedom, enabling AIC- or BIC-based hyperparameter selection for the "efron" model (Tan et al. 2025). See `pvEBayes_tune` for details.

Value

The function returns an S3 object of class `pvEBayes` containing the estimated model parameters as well as posterior draws for each AE-drug combination if the number of posterior draws is specified.

The `convergence` component is an integer code: `0` indicates successful convergence of the optimizer, while `1` indicates that the optimizer did not converge.

References

DuMouchel W. Bayesian data mining in large frequency tables, with an application to the FDA spontaneous reporting system. *The American Statistician.* 1999; 1;53(3):177-90.

Tan Y, Markatou M and Chakraborty S. Flexible Empirical Bayesian Approaches to Pharmacovigilance for Simultaneous Signal Detection and Signal Strength Estimation in Spontaneous Reporting Systems Data. *Statistics in Medicine.* 2025; 44: 18-19, https://doi.org/10.1002/sim.70195.

Narasimhan B, Efron B. deconvolveR: A G-modeling program for deconvolution and empirical Bayes estimation. *Journal of Statistical Software*. 2020; 2;94:1-20.

Koenker R, Gu J. REBayes: an R package for empirical Bayes mixture methods. *Journal of Statistical Software*. 2017; 4;82:1-26.

Fu, A, Narasimhan, B, Boyd, S. CVXR: An R Package for Disciplined Convex Optimization. *Journal of Statistical Software*. 2020; 94;14:1-34.

Examples

```
set.seed(1)

# fit general-gamma model with a specified alpha
fit <- pvEBayes(
```

```
  contin_table = statin2025_44,
  model = "general-gamma",
  alpha = 0.3,
  n_posterior_draws = 1000
)

# fit K-gamma model with K = 3
fit_Kgamma <- pvEBayes(
  contin_table = statin2025_44, model = "K-gamma",
  K = 3, n_posterior_draws = 1000
)


# fit Efron model with specified hyperparameters
# p = 40, c0 = 0.05

fit_efron <- pvEBayes(
  contin_table = statin2025_44,
  model = "efron",
  p = 40,
  c0 = 0.05,
  n_posterior_draws = 1000
)

# fit GPS model and comapre with 'openEBGM'


fit_gps <- pvEBayes(statin2025_44, model = "GPS")

## Not run:

## Optional comparison with openEBGM (only if installed)

## tol_ecm is the absolute tolerance for ECM stopping rule.
## It is set to ensure comparability to `openEBGM`.

fit_gps <- pvEBayes(statin2025_44, model = "GPS", tol_ecm = 1e-2)

if (requireNamespace("openEBGM", quietly = TRUE)) {
  E <- estimate_null_expected_count(statin2025_44)
  statin2025_44_stacked <- as.data.frame(as.table(statin2025_44))
  statin2025_44_stacked$E <- as.vector(E)
  colnames(statin2025_44_stacked) <- c("var1", "var2", "N", "E")
  statin2025_44_stacked_squash <- openEBGM::autoSquash(statin2025_44_stacked)

  hyper_estimates <- openEBGM::hyperEM(statin2025_44_stacked_squash,
    theta_init = c(2, 1, 2, 2, 0.2),
    method = "nlminb",
    N_star = NULL,
    zeroes = TRUE,
    param_upper = Inf,
    LL_tol = 1e-2,
    max_iter = 10000
```

```
  )
}

theta_hat <- hyper_estimates$estimates
qn <- openEBGM::Qn(theta_hat,
  N = statin2025_44_stacked$N,
  E = statin2025_44_stacked$E
)

statin2025_44_stacked$q05 <- openEBGM::quantBisect(5,
  theta_hat = theta_hat,
  N = statin2025_44_stacked$N,
  E = statin2025_44_stacked$E,
  qn = qn
)

## obtain the detected signal provided by openEBGM
statin2025_44_stacked %>%
  subset(q05 > 1.001)

## detected signal from pvEBayes presented in the same way as openEBGM
fit_gps %>%
  summary(return = "posterior draws") %>%
  apply(c(2, 3), quantile, prob = 0.05) %>%
  as.table() %>%
  as.data.frame() %>%
  subset(Freq > 1.001)

## End(Not run)
```

`pvEBayes_tune` *Select hyperparameter and obtain the optimal general-gamma or efron model based on AIC and BIC*

Description

This function performs hyperparameter tuning for the general-gamma or Efron model using AIC or BIC. For a given AE-drug contingency table, the function fits a series of models across a grid of candidate hyperparameter values and computes their AIC and BIC. The models with the lowest AIC or BIC values are selected as the optimal fits.

Usage

```
pvEBayes_tune(
  contin_table,
  model = c("general-gamma", "efron"),
  alpha_vec = NULL,
  p_vec = NULL,
```

```
  c0_vec = NULL,
  use_AIC = TRUE,
  n_posterior_draws = 1000,
  return_all_fit = FALSE,
  return_all_AIC = TRUE,
  return_all_BIC = TRUE,
  tol_ecm = 1e-04,
  rtol_efron = 1e-10,
  E = NULL
)
```

Arguments

contin_table	an IxJ contingency table showing pairwise counts of adverse events for I AEs (along the rows) and J drugs (along the columns).
model	the model to be tuned. Available models are "general-gamma" and "efron". Default to "general-gamma". Note that the input for model is case-sensitive.
alpha_vec	vector of hyperparameter alpha values to be selected. Alpha is a hyperparameter in general-gamma model which is numeric and between 0 and 1. If is NULL, a default set of alpha values (0, 0.1, 0.3, 0.5, 0.7, 0.9) will be used.
p_vec	vector of hyperparameter p values to be selected. p is a hyperparameter in "efron" model which should be a positive integer. If is NULL, a default set of p values (40, 60, 80, 100, 120) will be used.
c0_vec	vector of hyperparameter c0 values to be selected. c0 is a hyperparameter in "efron" model which should be a positive number. If is NULL, a default set of c0 values (0.001, 0.01, 0.1, 0.2, 0.5) will be used.
use_AIC	logical, indicating whether AIC or BIC is used. Default to be TRUE.
n_posterior_draws	number of posterior draws for each AE-drug combination.
return_all_fit	logical, indicating whether to return all the fitted model under the selection. Default to be FALSE.
return_all_AIC	logical, indicating whether to return AIC values for each fitted model under the selection. Default to be TRUE.
return_all_BIC	logical, indicating whether to return BIC values for each fitted model under the selection. Default to be TRUE.
tol_ecm	a tolerance parameter used for the ECM stopping rule, defined as the absolute change in the joint marginal likelihood between two consecutive iterations. It is used when 'GPS', 'K-gamma' or 'general-gamma' model is fitted. Default to be 1e-4.
rtol_efron	a tolerance parameter used when 'efron' model is fitted. Default to 1e-10. See 'stats::nlminb' for detail.
E	A matrix of expected counts under the null model for the SRS frequency table. If NULL (default), the expected counts are estimated from the SRS data using 'estimate_null_expected_count()'.

Value

The function returns an S3 object of class `pvEBayes` containing the selected estimated model parameters as well as posterior draws for each AE-drug combination if the number of posterior draws is specified.

References

Akaike H. A new look at the statistical model identification. *IEEE Transactions on Automatic Control.* 2003; 19(6):716-23.

Schwarz G. Estimating the dimension of a model. *The Annals of Statistics.* 1978; 1:461-4.

Tan Y, Markatou M and Chakraborty S. Flexible Empirical Bayesian Approaches to Pharmacovigilance for Simultaneous Signal Detection and Signal Strength Estimation in Spontaneous Reporting Systems Data. *Statistics in Medicine.* 2025; 44: 18-19, https://doi.org/10.1002/sim.70195.

Examples

```
fit <- pvEBayes_tune(statin2025_44,
  model = "general-gamma",
  alpha_vec = c(0, 0.1, 0.3, 0.5, 0.7, 0.9)
)
```

---

statin2025 *FDA statin dataset with 5119 adverse events*

Description

An adverse event-drug count dataset (contingency table) obtained from the FDA FAERS database for the quarters 2021Q1 - 2024Q4.

Usage

```
statin2025
```

Format

An object of class `matrix` (inherits from `array`) with 5119 rows and 7 columns.

Details

The dataset catalogs 6 statin drugs (across columns): Atorvastatin, Fluvastatin, Lovastatin, Pravastatin, Rosuvastatin and Simvastatin.

Data are stored in the form of a contingency table, with drugs listed across the columns and the 5119 AEs presented across the rows. Each cell in the contingency table represents the total report counts associated with that (AE, drug) pair and detected in the FDA FAERS database during 2021Q1 - 2024Q4.

The dataset catalogs 6 statin drugs (across columns):

Atorvastatin, Fluvastatin, Lovastatin, Pravastatin, Rosuvastatin and Simvastatin.

The 5119 adverse events presented across the rows are AEs that contain at least one report for 6 statin drugs during 2021Q1 - 2024Q4.

This dataset is an updated version of statin from the pvLRT package which collects the same scope of AEs for 6 statin drugs for quarters 2014Q3 - 2020Q4.

Source

https://fis.fda.gov/extensions/FPD-QDE-FAERS/FPD-QDE-FAERS.html

---

`statin2025_44` *FDA statin dataset with 44 adverse events*

Description

An adverse event-drug count dataset (contingency table) obtained from the FDA FAERS database for the quarters 2021Q1 - 2024Q4.

Usage

```
statin2025_44
```

Format

An object of class `matrix` (inherits from `array`) with 45 rows and 7 columns.

Details

Data are stored in the form of a contingency table, with drugs listed across the columns and the 44 AEs presented across the rows. Each cell in the contingency table represents the total report counts associated with that (AE, drug) pair and detected in the FDA FAERS database during 2021Q1 - 2024Q4.

The dataset catalogs 6 statin drugs (across columns):

Atorvastatin, Fluvastatin, Lovastatin, Pravastatin, Rosuvastatin and Simvastatin.

The 44 adverse events presented across the rows are considered significant by FDA.

This dataset is an updated version of statin46 from the pvLRT package which collect the same scope of AEs for 6 statin drugs for quarters 2014Q3 - 2020Q4.

During 2021Q1 - 2024Q4, there was no AE report for "BLOOD CREATINE PHOSPHOKINASE MM INCREASED" and "MYOGLOBIN BLOOD PRESENT". Therefore, these two AEs are not presented in the statin2025_44 dataset.

Source

https://fis.fda.gov/extensions/FPD-QDE-FAERS/FPD-QDE-FAERS.html

statin42 *FDA statin dataset with 42 adverse events*

Description

An adverse event-drug count dataset (contingency table) obtained from the FDA FAERS database for the quarters 2014Q3 - 2020Q4.

Usage

```
statin42
```

Format

An object of class `matrix` (inherits from `array`) with 43 rows and 7 columns.

Details

Data are stored in the form of a contingency table, with drugs listed across the columns and the 42 AEs presented across the rows. Each cell in the contingency table represents the total report counts associated with that (AE, drug) pair and detected in the FDA FAERS database during 2014Q3 - 2020Q4.

The dataset catalogs 6 statin drugs (across columns):

Atorvastatin, Fluvastatin, Lovastatin, Pravastatin, Rosuvastatin and Simvastatin.

This dataset is derived from the `statin46` dataset in the **pvLRT** package, with four AEs removed.

The excluded AEs are: "Blood Creatine Phosphokinase Mm Increased", "Myoglobin Blood Present", "Myoglobin Urine Present", and "Myoglobinaemia".

Source

https://fis.fda.gov/extensions/FPD-QDE-FAERS/FPD-QDE-FAERS.html

summary.pvEBayes *Summary method for a pvEBayes object*

Description

This function defines the S3 `summary` method for objects of class `pvEBayes`. It provides a detailed summary of the fitted model.

Usage

```
## S3 method for class 'pvEBayes'
summary(object, return = NULL, ...)
```

Arguments

`object`	a `pvEBayes` object, which is the output of the function pvEBayes or pvEBayes_tune.
`return`	a character string specifying which component the summary function should return.Valid options include: "prior parameters", "likelihood", "detected signal", "posterior draws" and "posterior draws long format". If set to NULL (default), a summary table will be returned (see 'summary_table_pvEBayes()'). Note that the input for 'return' is case-sensitive.
`...`	other input parameters. Currently unused.

Value

If `return = NULL` (default), the function returns a summary table generated by `summary_table_pvEBayes()`, after printing the fitted `pvEBayes` object.

If `return` is specified, the function returns the requested component:

`prior parameters` A list of estimated prior parameters.

`likelihood` The fitted model log marginal likelihood.

`detected signal` A logical matrix indicating AE-drug pairs if $P(\lambda > 1.001 \mid N) > 0.95$. For signal detection with specified threshold parameters, see 'get_posterior_prob()'

`posterior draws` Posterior draws of the signal strength for each AE-drug pair in default array format.

`posterior draws long format` Posterior draws of the signal strength for each AE-drug pair in stacked long format.

Examples

```
obj <- pvEBayes(
  contin_table = statin2025_44, model = "general-gamma",
  alpha = 0.5, n_posterior_draws = 10000
)

summary(obj)
```

---

summary_table_pvEBayes

Obtain a summary table for a pvEBayes object

Description

Obtain a summary table for a pvEBayes object

Usage

```
summary_table_pvEBayes(x, cutoff_signal = 1.001)
```

Arguments

x	a pvEBayes object, which is the output of the function pvEBayes or pvEBayes_tune.
cutoff_signal	numeric. Threshold for signal detection. An AE-drug combination is classified as a detected signal if its 5th posterior percentile exceeds this threshold.

Value

a data.table that summarizes reporting count (N), expected null value (E), posterior probability of being a signal (post_prob), posterior signal strength median (q50), 5-th and 95-th posterior signal strength percentile (q05 and q95) for each AE-drug combination.

Examples

```
fit <- pvEBayes(
  contin_table = statin2025_44, model = "general-gamma",
  alpha = 0.5, n_posterior_draws = 100
)

summary_table_pvEBayes(fit)
```

---

vigi_opioid_mental *VigiBase opioid dataset with 100 mental-related adverse events*

Description

An adverse event-drug count dataset (contingency table) obtained from VigiBase.

Usage

```
vigi_opioid_mental
```

Format

An object of class matrix (inherits from array) with 101 rows and 6 columns.

Details

Data are stored in the form of a contingency table, with drugs listed across the columns and the 100 AEs presented across the rows. Each cell in the contingency table represents the total report counts associated with that (AE, drug) pair and detected in VigiBase.

The dataset catalogs 5 opioid drugs (across columns):

Codeine, Fentanyl, Oxycodone, Pentazocine and Tramadol.

Source

https://www.vigiaccess.org/

References

Tan Y, Markatou M and Chakraborty S. A Review of Statistical Methods for Spontaneous Reporting System Data Mining: Signal Detection and Beyond. *arXiv*:2604.18898 (stat.AP). https://doi.org/10.48550/arXiv.2604.18898

Index