Package ‘vacalibration’

July 24, 2025

Title Calibration of Computer-Coded Verbal Autopsy Algorithm

Version 2.0

Maintainer Sandipan Pramanik <`sandy.pramanik@gmail.com`>

Description Calibrates cause-specific mortality fractions (CSMF) estimates generated by computer-coded verbal autopsy (CCVA) algorithms from WHO-standardized verbal autopsy (VA) survey data. It leverages data from the multi-country Child Health and Mortality Prevention Surveillance (CHAMPS) project <`https://champshealth.org/`>, which determines gold standard causes of death via Minimally Invasive Tissue Sampling (MITS). By modeling the CHAMPS data using the misclassification matrix modeling framework proposed in Pramanik et al. (2025, <doi:10.1214/24-AOAS2006>), the package includes an inventory of 48 uncertainty-quantified misclassification matrices for three CCVA algorithms (EAVA, InSilicoVA, InterVA), two age groups (neonates aged 0-27 days and children aged 1-59 months), and eight ``countries'' (seven countries in CHAMPS -- Bangladesh, Ethiopia, Kenya, Mali, Mozambique, Sierra Leone, South Africa -- and an estimate for countries not in CHAMPS). Given a VA-only data for an age group, CCVA algorithm, and country, the package uses the corresponding uncertainty-quantified misclassification matrix estimates as an informative prior, and utilizes the modular VA-calibration to produce calibrated CSMF estimates. It also supports ensemble calibration when VA-only data are provided for multiple algorithms. More generally, the package can be applied to calibrate predictions from a discrete classifier (or ensemble of classifiers) utilizing user-provided fixed or uncertainty-quantified misclassification matrices. This work is supported by the Bill and Melinda Gates Foundation Grant INV-034842.

License GPL-2

Encoding UTF-8

RoxygenNote 7.3.2

Imports rstan, ggplot2, loo, patchwork, reshape2

Config/testthat/edition 3

Depends R (>= 3.5)

LazyData true

Suggests knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation no

Author Sandipan Pramanik [aut, cre] (ORCID:
<https://orcid.org/0000-0002-7196-155X>),
Emily Wilson [aut],
Jacob Fiksel [aut],
Brian Gilbert [aut],
Abhirup Datta [aut]

Repository CRAN

Date/Publication 2025-07-24 12:50:02 UTC

Contents

`cause_map` *Broad Cause Mapping*

Description

Maps individual-level specific (high resolution) cause of death (`codEAVA()` function in `EAVA` and `crossVA()` function in `openVA`) to broad causes.

Usage

```
cause_map(df, age_group)
```

Arguments

`df`	Data frame. Outputs from `crossVA()` function in `openVA` for EAVA and `crossVA()` function in `openVA` for InSilicoVA and InterVA
`age_group`	Character. The age group of interest. "neonate" for deaths between 0-27 days, and "child" for 1-59 months.

Value

Matrix. Rows are individuals. Columns are broad causes. This is a binary matrix (entries 0 or 1) with 1 indicating the broad cause of death for the individual.

Examples

```
## COMSA-Mozambique Publicly Available Version
## Example Individual-Level Specific (High-Resolution) Cause of Death Data
data(comsamoz_public_openVAout)
head(comsamoz_public_openVAout$data)  # head of the data
comsamoz_public_openVAout$data[1,]  # ID and specific cause of death for individual 1

## mapped to broad cause
## same as comsamoz_public_broad$data
comsamoz_public_asbroad = cause_map(df = comsamoz_public_openVAout$data, age_group = "neonate")
head(comsamoz_public_asbroad)

### store broad cause map of the data
data(comsamoz_public_broad)
head(comsamoz_public_broad$data) # identical to head(comsamoz_public_asbroad)
```

`comsamoz_public_broad` *COMSA-Mozambique: Example Individual-Level Broad Cause of Death Data (Publicly Available Version)*

Description

Example individual-level neonatal cause-of-death data using InSilicoVA. This is obtained after broad cause mapping of `comsamoz_public_openVAout$data` using `cause_map()` function in this package.

Usage

```
comsamoz_public_broad
```

Format

A list of 4 components.

data Binary matrix. Contains the data. Rows are individuals. Columns are broad causes. Matrix elements are 0 or 1, with 1 indicating the cause of death for an individual.

age_group Character. Indicate age group. "neonate" (for 0-27 days) for this data

va_algo Character. Indicate CCVA algorithm. "insilicova" for this data

version Character. Date stamp for version control of tracking updates. Only for package maintainers.

Details

This shows how individual level broad cause of death data can be an input in the `vacalibration()` function for calibration.

`comsamoz_public_broad$data[i,j]` is a binary indicator of whether broad cause `j` is the cause of death for individual `i`. 1 indicates it is, and 0 indicates it is not.

Broad causes for "neonate" are

- "congenital_malformation",
- "pneumonia",
- "sepsis_meningitis_inf" (sepsis/meningitis/infections),
- "ipre" (intrapartum-related events),
- "other", and
- "prematurity".

For "child", the broad causes are

- "malaria",
- "pneumonia",
- "diarrhea",
- "severe_malnutrition",
- "hiv",
- "injury",
- "other",
- "other_infections", and
- "nn_causes" (neonatal causes; consists of IPRE, congenital malformation, and prematurity).

References

Macicame, I, et al. (2023). *Countrywide Mortality Surveillance for Action in Mozambique: Results from a National Sample-Based Vital Statistics System for Mortality and Cause of Death.* American Journal of Tropical Medicine and Hygiene, 108(Suppl 5), pp. 5–16.

Examples

```
## using the data
data(comsamoz_public_broad)
head(comsamoz_public_broad$data)  # head of the data
comsamoz_public_broad$data[1,]  # binary vector indicating cause of death for individual 1

## mapped to national death counts
comsamoz_public_asdeathcount = colSums(comsamoz_public_broad$data)

## VA-calibration for the "neonate" age group and InSilicoVA algorithm
## input as broad cause
```

```
calib_out_asbroad = vacalibration(va_data = setNames(list(comsamoz_public_broad$data),
                                              list(comsamoz_public_broad$va_algo)),
                                  age_group = comsamoz_public_broad$age_group,
                                  country = "Mozambique")

## input as specific cause
calib_out_asdeathcount = vacalibration(va_data = setNames(list(comsamoz_public_asdeathcount),
                                              list(comsamoz_public_broad$va_algo)),
                                   age_group = comsamoz_public_broad$age_group,
                                   country = "Mozambique")

## comparing uncalibrated CSMF estimates and posterior summary of calibrated CSMF estimates
## all are the same
calib_out_asbroad$p_uncalib
calib_out_asbroad$pcalib_postsumm[1,,]

calib_out_asdeathcount$p_uncalib
calib_out_asdeathcount$pcalib_postsumm[1,,]
```

`comsamoz_public_openVAout` *COMSA-Mozambique: Example Individual-Level Specific (High-Resolution) Cause of Death Data (Publicly Available Version)*

Description

Example individual-level neonatal cause-of-death data using InSilicoVA. This is obtained by applying InSilicoVA algorithm and `crossVA` mapping in the `openVA` package. This provides specific (high-resolution) cause of death for each individual.

Usage

```
comsamoz_public_openVAout
```

Format

A list of 4 components.

- **data** Data frame. Contains the data. Rows are individuals. It has 2 columns. First column "ID" is the individual ID. Second column "cause" are the high-resolution causes of deaths.
- **age_group** Character. Indicate age group. "neonate" (for 0-27 days) for this data
- **va_algo** Character. Indicate CCVA algorithm. "insilicova" for this data
- **version** Character. Date stamp for version control of tracking updates. Only for package maintainers.

Details

`comsamoz_public_openVAout$data$ID[i]` is the ID for individual `i`.

`comsamoz_public_openVAout$data$cause[i]` is the specific cause of death for individual `i`.

References

Macicame, I, et al. (2023). *Countrywide Mortality Surveillance for Action in Mozambique: Results from a National Sample-Based Vital Statistics System for Mortality and Cause of Death.* American Journal of Tropical Medicine and Hygiene, 108(Suppl 5), pp. 5–16.

Examples

```
## using the data (as output by crossVA function in openVA package for InSilicoVA algorithm)
data(comsamoz_public_openVAout)
head(comsamoz_public_openVAout$data)  # head of the data
comsamoz_public_openVAout$data[1,]  # ID and specific cause of death for individual 1

## mapped to broad cause
### same as comsamoz_public_broad$data
comsamoz_public_asbroad = cause_map(df = comsamoz_public_openVAout$data, age_group = "neonate")
head(comsamoz_public_asbroad)

### store broad cause map of the data
data(comsamoz_public_broad)
head(comsamoz_public_broad$data) # identical to head(comsamoz_public_asbroad)

## mapped to national death counts
comsamoz_public_asdeathcount = colSums(comsamoz_public_asbroad)

## VA-calibration for the "neonate" age group and InSilicoVA algorithm
## input as specific cause
calib_out_asspecific = vacalibration(va_data = setNames(list(comsamoz_public_openVAout$data),
                                              list(comsamoz_public_openVAout$va_algo)),
                                     age_group = comsamoz_public_openVAout$age_group,
                                     country = "Mozambique")

## input as broad cause
calib_out_asbroad = vacalibration(va_data = setNames(list(comsamoz_public_asbroad),
                                              list(comsamoz_public_openVAout$va_algo)),
                                     age_group = comsamoz_public_openVAout$age_group,
                                     country = "Mozambique")

## input as specific cause
calib_out_asdeathcount = vacalibration(va_data = setNames(list(comsamoz_public_asdeathcount),
                                              list(comsamoz_public_openVAout$va_algo)),
                                       age_group = comsamoz_public_openVAout$age_group,
                                       country = "Mozambique")

## comparing uncalibrated CSMF estimates and posterior summary of calibrated CSMF estimates
```

```
calib_out_asspecific$p_uncalib
calib_out_asspecific$pcalib_postsumm[1,,]

calib_out_asbroad$p_uncalib
calib_out_asbroad$pcalib_postsumm[1,,]

calib_out_asdeathcount$p_uncalib
calib_out_asdeathcount$pcalib_postsumm[1,,]
```

`Mmat_champs` *Misclassification Estimates Based on CHAMPS Data*

Description

Estimates of misclassification matrices using the modeling framework from Pramanik et al. (2025) and the limited paired MITS-VA data from the Child Health and Mortality Prevention Surveillance (CHAMPS) project.

Usage

```
Mmat_champs
```

Format

A nested list.

age_group "neonate" for 0-27 days, and "child" for 1-59 months

va_algo "eava", "insilicova", and "interva"

estimate types "postsumm" contains posterior summaries, "postmean" contains the posterior means, and "asDirich" contains Dirichlet approximation for each CHAMPS cause and country.

country "Bangladesh", "Ethiopia", "Kenya", "Mali", "Mozambique", "Sierra Leone", "South Africa", "other"

version Date stamp for version control of tracking updates. Only for package maintainers.

Details

`Mmat_champs[[age_group]][[va_algo]][["postsumm"]][[country]]` contains posterior summaries of misclassification matrix for the a desired age_group, va_algo, and country. It is an array of dimension the number of posterior summaries X CHAMPS broad cause X VA broad cause. For example, if analyzing "neonate" age group using "insilicova" algorithm in "Mozambique",

- `Mmat_champs$neonate$insilicova$postsumm$Mozambique[,"pneumonia","pneumonia"]` are posterior summaries of the sensitivity for "pneumonia".

- `Mmat_champs$neonate$insilicova$postsumm$Mozambique[,"pneumonia","ipre"]` are posterior summaries of the false negative rate for CHAMPS broad cause "pneumonia" and VA broad cause "ipre".

Posterior samples are available from the GitHub repository https://github.com/sandy-pramanik/Mmat_champs.

.rda file is available under the release: https://github.com/sandy-pramanik/Mmat_champs/releases/tag/20241004.

`Mmat_champs[[age_group]][[va_algo]][["postmean"]][[country]]` contains posterior means.

`Mmat_champs[[age_group]][[va_algo]][["asDirich"]][[country]]` contains Dirichlet approximations of its posterior.

They are matrices of dimension CHAMPS broad cause X VA broad cause. For example, if analyzing "neonate" age group using "insilicova" algorithm in "Mozambique",

- `Mmat_champs$neonate$insilicova$postmean$Mozambique["pneumonia","pneumonia"]` is the posterior mean of sensitivity for "pneumonia".
- `Mmat_champs$neonate$insilicova$postmean$Mozambique["pneumonia","ipre"]` is the posterior mean of false negative rate for CHAMPS broad cause "pneumonia" and VA broad cause "ipre".

Similarly, `Mmat_champs$neonate$insilicova$asDirich$Mozambique["pneumonia",]` are parameters of Dirichlet distribution approximating the posterior of classification rates of different broad causes for the CHAMPS broad cause "pneumonia".

References

Pramanik, S, et al. (2025). *Modeling structure and country-specific heterogeneity in misclassification matrices of verbal autopsy-based cause of death classifiers*. Annals of Applied Statistics, 19(2):1214–1239. ISSN 1932-6157.

Taylor, A, et al. (2020). *Initial findings from a novel population-based child mortality surveillance approach: a descriptive study*. Lancet Glob Health, 8(7):e909-e919.

Examples

```
## misclassification estimates
data(Mmat_champs)

# misclassification estimates for "neonate" age group and "insilicova" algorithm in Mozambique
## posterior summaries of the sensitivity of "pneumonia"
Mmat_champs$neonate$insilicova$postsumm$Mozambique[,"pneumonia","pneumonia"]

## posterior summaries of the false negative rates
## CHAMPS cause "pneumonia" and VA cause "ipre"
Mmat_champs$neonate$insilicova$postsumm$Mozambique[,"pneumonia","ipre"]

# COMSA-Mozambique: Example (Publicly Available Version)
# Individual-Level Specific (High-Resolution) Cause of Death Data
```

```
data(comsamoz_public_openVAout)
head(comsamoz_public_openVAout$data)  # head of the data

## VA-calibration for the "neonate" age group and "insilicova" algorithm
calib_out1 = vacalibration(va_data =
                                     setNames(list(comsamoz_public_openVAout$data),
                                              list(comsamoz_public_openVAout$va_algo)),
                           age_group = comsamoz_public_openVAout$age_group,
                           country = "Mozambique")

calib_out2 = vacalibration(va_data =
                                     setNames(list(comsamoz_public_openVAout$data),
                                              list(comsamoz_public_openVAout$va_algo)),
                           age_group = comsamoz_public_openVAout$age_group,
                           country = "Mozambique",
 Mmat.asDirich = list("insilicova" = Mmat_champs$neonate$insilicova$asDirich$Mozambique))
## By default the function fetches the desired misclassification estimates from
## the stored Mmat_champs.

## So calib_out1 (where we don't specify the misclassification) and
## calib_out2 (where we specify) are identical.
```

`modular.vacalib` *Modular VA-Calibration*

Description

Modular VA-Calibration

Usage

```
modular.vacalib(
  va_unlabeled = NULL,
  age_group = NULL,
  calibmodel.type = c("Mmatprior", "Mmatfixed")[1],
  Mmat.asDirich = NULL,
  Mmat.fixed = NULL,
  donotcalib = NULL,
  donot.calib_type = c("learn", "fixed")[1],
  nocalib.threshold = 0.1,
  stable = TRUE,
  ensemble = NULL,
  pss = NULL,
  nMCMC = 5000,
  nBurn = 5000,
  nThin = 1,
```

```
  adapt_delta_stan = 0.9,
  refresh.stan = NULL,
  seed = 1,
  verbose = TRUE,
  saveoutput = FALSE,
  output_filename = NULL,
  plot_it = TRUE
)
```

Arguments

`va_unlabeled`	A named list. Algorithm-specific unlabeled VA-only data. For example, `list("algo1" = algo1_output, "algo2" = algo2_output, ...)` Algorithm names (`"algo1"`, `"algo2"`, ...) can be "eava", "insilicova", or "interva". Data (`algo1_output`, `algo2_output`, ...) can be broad causes (output from the `cause_map()` function in this package), or broad-cause-specific death counts (integer vector). Can be different for different algorithms. Total number of deaths for different algorithms can be different.
`age_group`	Character. Age-group of interest. `"neonate"` or `"child"`. `"neonate"` ages between 0-27 days, or `"child"` ages between 1-59 months.
`calibmodel.type`	Character. How to utilize misclassification estimates. `"Mmatprior"` (default). Propagates uncertainty in the misclassification matrix estimates. `"Mmatfixed"`. Uses fixed (default: posterior mean) misclassification matrix estimates.
`Mmat.asDirich`	A named list. Similarly structured as `va_data`. Needed only if `calibmodel.type = "Mmatprior"` (propagates uncertainty). For example, `list("algo1" = Mmat.asDirich_algo1, "algo2" = Mmat.asDirich_algo2, ...)`. List of algorithm-specific Dirichlet prior on misclassification matrix to be used for calibration. Names and length must be identical to `va_data`. If algorithm names (`"algo1"`, `"algo2"`, ...) are `"eava"`, `"insilicova"` or `"interva"`, and `Mmat.asDirich` is missing, it by default uses the CHAMPS-based estimates (Dirichlet approximation of posterior) stored in `Mmat_champs` in this package. See `Mmat_champs` for details. If `Mmat.asDirich` is not missing, whatever provided is used. If any algorithm name (`"algo1"`, `"algo2"`, ...) is different from `"eava"`, `"insilicova"` or `"interva"`, `Mmat.asDirich` must be provided. `Mmat.asDirich_algo1` is a matrix of dimension CHAMPS ("gold standard") cause by VA cause.

	`Dirichlet(Mmat.asDirich_algo1[i,])` is used as informative prior on classification rates for CHAMPS cause `i`.
`Mmat.fixed`	A named list. Similarly structured as `va_data` or `Mmat.asDirich`. Needed only if `calibmodel.type = "Mmatfixed"` (no uncertainty propagation). For example, `list("algo1" = Mmat.fixed_algo1, "algo2" = Mmat.fixed_algo2, ...)` List of algorithm-specific fixed misclassification matrix to be used for calibration. Names and length must be identical to `va_data`. If algorithm names (`"algo1"`, `"algo2"`, ...) are `"eava"`, `"insilicova"`, or `"interva"` and `Mmat.fixed` is missing, it by default uses the CHAMPS-based estimates (posterior mean) stored in `Mmat_champs` in this package. See `Mmat_champs` for details. If `Mmat.fixed` is not missing, whatever provided is used. If any algorithm name (`"algo1"`, `"algo2"`, ...) is different from `"eava"`, `"insilicova"` or `"interva"`, `Mmat.fixed` must be provided. `Mmat.fixed_algo1` is a matrix of dimension CHAMPS cause X VA cause. `Mmat.fixed_algo1[i,]` are the classification rates for CHAMPS cause `i`.
`donotcalib`	A named list. Similarly structured as `va_data`, `Mmat.asDirich`, or `Mmat.fixed`. List of broad causes for each CCVA algorithm that we do not want to calibrate Default: `list("eava"="other", "insilicova"="other", "interva"="other")`. That is, `"other"` cause is not calibrated. For neonates, the broad causes are `"congenital_malformation"`, `"pneumonia"`, `"sepsis_meningitis_inf"`, `"ipre"`, `"other"`, or `"prematurity"`. For children, the broad causes are `"malaria"`, `"pneumonia"`, `"diarrhea"`, `"severe_malnutrition"`, `"hiv"`, `"injury"`, `"other"`, `"other_infections"`, `"nn_causes"` (neonatal causes). Set `list("eava" = NULL, "insilicova" = NULL, "interva" = NULL)` if you want to calibrate all causes.
`donot.calib_type`	Character. `"fixed"` or `"learn"` (default). For `"fixed"`, only broad causes that are provided in `"donotcalib"` are not calibrated. For `"learn"`, it learns from `"Mmat.fixed"` or `"Mmat.asDirich"` if any other causes cannot be calibrated. For `"learn"`, it identifies VA causes for which the misclassification rates do not vary across CHAMPS causes. In that case, the calibration equation becomes ill-conditioned (see the footnote below Section 3.8 in Pramanik et al. (2025)). Currently, we address this by not calibrating VA causes for which the misclassification rates are similar along the rows (CHAMPS causes). VA causes (Columns) for which the rates along the rows (CHAMPS causes) do not vary more that `"nocalib.threshold"` are not calibrated. `"donotcalib"` is accordingly updated for each CCVA algorithm.
`nocalib.threshold`	Numeric between 0 and 1. The value used for screening VA causes that cannot be calibrated when `donot.calib_type = "learn"`. Default: 0.1.

`stable`	Logical. `TRUE` (default) or `FALSE`. Setting `TRUE` improves stability in calibration.
`ensemble`	Logical. `TRUE` (default) or `FALSE`. Whether to perform ensemble calibration when outputs from multiple algorithms are provided.
`pss`	Positive numeric. Degree of shrinkage of calibrated cause-specific mortality fraction (CSMF) estimate towards uncalibrated estimates. Always 0 when `stable=TRUE`. Defaults to 4 when `stable=FALSE`.
`nMCMC`	Positive integer. Total number of posterior samples to perform inference on. Total number of iterations are `nBurn + nMCMC*nThin`. Default 5000.
`nBurn`	Positive integer. Total burn-in in posterior sampling. Total number of iterations are `nBurn + nMCMC*nThin`. Default 5000.
`nThin`	Positive integer. Number of thinning in posterior sampling. Total number of iterations are `nBurn + nMCMC*nThin`. Default 1.
`adapt_delta_stan`	Positive numeric between 0 and 1. `"adapt_delta"` parameter in `rstan`. Influences the behavior of the No-U-Turn Sampler (NUTS), the primary MCMC sampling algorithm in Stan. Default 0.9.
`refresh.stan`	Positive integer. Report progress at every `refresh.stan`-th iteration. Default `(nBurn + nMCMC*nThin)/10`, that is at every 10% progress.
`seed`	Numeric. `"seed"` parameter in rstan. Default 1.
`verbose`	Logical. Reports progress or not. `TRUE` (default) or `FALSE`.
`saveoutput`	Logical. Save output or not. `TRUE` (default) or `FALSE`.
`output_filename`	Character. Output name to save as. Default `paste0("calibratedva_", calibmodel.type)`. That is `"calibratedva_Mmatprior"` or `"calibratedva_Mmatfixed"`.
`plot_it`	Logical. Whether to return comparison plot for summary. `TRUE` (default) or `FALSE`.

Value

A named list. Use `vacalibration()` for general purpose.

`vacalibration` *VA-calibration function*

Description

VA-calibration function

Usage

```
vacalibration(
  va_data = NULL,
  age_group = NULL,
  country = NULL,
  calibmodel.type = c("Mmatprior", "Mmatfixed")[1],
  Mmat.asDirich = NULL,
  Mmat.fixed = NULL,
  donotcalib = NULL,
  donot.calib_type = c("learn", "fixed")[1],
  nocalib.threshold = 0.1,
  stable = TRUE,
  ensemble = NULL,
  pss = NULL,
  nMCMC = 5000,
  nBurn = 5000,
  nThin = 1,
  adapt_delta_stan = 0.9,
  refresh.stan = NULL,
  seed = 1,
  verbose = TRUE,
  saveoutput = FALSE,
  output_filename = NULL,
  plot_it = TRUE
)
```

Arguments

`va_data`	A named list. Algorithm-specific unlabeled VA-only data. For example, `list("algo1" = algo1_output, "algo2" = algo2_output, ...)`. Algorithm names (`"algo1"`, `"algo2"`, ...) can be "eava", "insilicova", or "interva". Data (`algo1_output`, `algo2_output`, ...) can be specific causes (output from `codEAVA()` function in `EAVA` and `crossVA()` function in `openVA`), or broad causes (output from the `cause_map()` function in this package), or broad-cause-specific death counts (integer vector). Can be different for different algorithms. Total number of deaths for different algorithms can be different.

`age_group`	Character. Age-group of interest. `"neonate"` or `"child"`. `"neonate"` ages between 0-27 days, or `"child"` ages between 1-59 months.
`country`	Character. The country `va_data` is from. Country-specific calibration is possible for "Bangladesh", "Ethiopia", "Kenya", "Mali", "Mozambique", "Sierra Leone", "South Africa". Any other country is matched with "other".
`calibmodel.type`	Character. How to utilize misclassification estimates. `"Mmatprior"` (default). Propagates uncertainty in the misclassification matrix estimates. `"Mmatfixed"`. Uses fixed (default: posterior mean) misclassification matrix estimates.
`Mmat.asDirich`	A named list. Similarly structured as `va_data`. Needed only if `calibmodel.type = "Mmatprior"` (propagates uncertainty). For example, `list("algo1" = Mmat.asDirich_algo1, "algo2" = Mmat.asDirich_algo2, ...)`. List of algorithm-specific Dirichlet prior on misclassification matrix to be used for calibration. Names and length must be identical to `va_data`. If algorithm names (`"algo1"`, `"algo2"`, ...) are `"eava"`, `"insilicova"` or `"interva"`, and `Mmat.asDirich` is missing, it by default uses the CHAMPS-based estimates (Dirichlet approximation of posterior) stored in `Mmat_champs` in this package. See `Mmat_champs` for details. If `Mmat.asDirich` is not missing, whatever provided is used. If any algorithm name (`"algo1"`, `"algo2"`, ...) is different from `"eava"`, `"insilicova"` or `"interva"`, `Mmat.asDirich` must be provided. `Mmat.asDirich_algo1` is a matrix of dimension CHAMPS ("gold standard") cause X VA cause. `Dirichlet(Mmat.asDirich_algo1[i,])` is used as informative prior on classification rates for CHAMPS cause `i`.
`Mmat.fixed`	A named list. Similarly structured as `va_data` or `Mmat.asDirich`. Needed only if `calibmodel.type = "Mmatfixed"` (no uncertainty propagation). For example, `list("algo1" = Mmat.fixed_algo1, "algo2" = Mmat.fixed_algo2, ...)` List of algorithm-specific fixed misclassification matrix to be used for calibration. Names and length must be identical to `va_data`. If algorithm names (`"algo1"`, `"algo2"`, ...) are `"eava"`, `"insilicova"`, or `"interva"` and `Mmat.fixed` is missing, it by default uses the CHAMPS-based estimates (posterior mean) stored in `Mmat_champs` in this package. See `Mmat_champs` for details. If `Mmat.fixed` is not missing, whatever provided is used.

	If any algorithm name (`"algo1"`, `"algo2"`, ...) is different from `"eava"`, `"insilicova"` or `"interva"`, `Mmat.fixed` must be provided. `Mmat.fixed_algo1` is a matrix of dimension CHAMPS cause X VA cause. `Mmat.fixed_algo1[i,]` are the classification rates for CHAMPS cause `i`.
`donotcalib`	A named list. Similarly structured as `va_data`, `Mmat.asDirich`, or `Mmat.fixed`. List of broad causes for each CCVA algorithm that we do not want to calibrate Default: `list("eava"="other", "insilicova"="other", "interva"="other")`. That is, `"other"` cause is not calibrated. For neonates, the broad causes are `"congenital_malformation"`, `"pneumonia"`, `"sepsis_meningitis_inf"`, `"ipre"`, `"other"`, or `"prematurity"`. For children, the broad causes are `"malaria"`, `"pneumonia"`, `"diarrhea"`, `"severe_malnutrition"`, `"hiv"`, `"injury"`, `"other"`, `"other_infections"`, `"nn_causes"` (neonatal causes). Set `list("eava" = NULL, "insilicova" = NULL, "interva" = NULL)` if you want to calibrate all causes.
`donot.calib_type`	Character. `"fixed"` or `"learn"` (default). For `"fixed"`, only broad causes that are provided in `"donotcalib"` are not calibrated. For `"learn"`, it learns from `"Mmat.fixed"` or `"Mmat.asDirich"` if any other causes cannot be calibrated. For `"learn"`, it identifies VA causes for which the misclassification rates do not vary across CHAMPS causes. In that case, the calibration equation becomes ill-conditioned (see the footnote below Section 3.8 in Pramanik et al. (2025)). Currently, we address this by not calibrating VA causes for which the misclassification rates are similar along the rows (CHAMPS causes). VA causes (Columns) for which the rates along the rows (CHAMPS causes) do not vary more that `"nocalib.threshold"` are not calibrated. `"donotcalib"` is accordingly updated for each CCVA algorithm.
`nocalib.threshold`	Numeric between 0 and 1. The value used for screening VA causes that cannot be calibrated when `donot.calib_type = "learn"`. Default: 0.1.
`stable`	Logical. `TRUE` (default) or `FALSE`. Setting `TRUE` improves stability in calibration.
`ensemble`	Logical. `TRUE` (default) or `FALSE`. Whether to perform ensemble calibration when outputs from multiple algorithms are provided.
`pss`	Positive numeric. Degree of shrinkage of calibrated cause-specific mortality fraction (CSMF) estimate towards uncalibrated estimates. Always 0 when `stable=TRUE`. Defaults to 4 when `stable=FALSE`.
`nMCMC`	Positive integer. Total number of posterior samples to perform inference on. Total number of iterations are `nBurn + nMCMC*nThin`. Default 5000.
`nBurn`	Positive integer. Total burn-in in posterior sampling. Total number of iterations are `nBurn + nMCMC*nThin`. Default 5000.

`nThin`	Positive integer. Number of thinning in posterior sampling. Total number of iterations are `nBurn + nMCMC*nThin`. Default 1.
`adapt_delta_stan`	Positive numeric between 0 and 1. `"adapt_delta"` parameter in `rstan`. Influences the behavior of the No-U-Turn Sampler (NUTS), the primary MCMC sampling algorithm in Stan. Default 0.9.
`refresh.stan`	Positive integer. Report progress at every `refresh.stan`-th iteration. Default `(nBurn + nMCMC*nThin)/10`, that is at every 10% progress.
`seed`	Numeric. `"seed"` parameter in rstan. Default 1.
`verbose`	Logical. Reports progress or not. `TRUE` (default) or `FALSE`.
`saveoutput`	Logical. Save output or not. `TRUE` (default) or `FALSE`.
`output_filename`	Character. Output name to save as. Default `paste0("calibratedva_", calibmodel.type)`. That is `"calibratedva_Mmatprior"` or `"calibratedva_Mmatfixed"`.
`plot_it`	Logical. Whether to return comparison plot for summary. `TRUE` (default) or `FALSE`.

Value

A named list:

input A named list of input data

p_uncalib Uncalibrated cause-specific mortality fractions (CSMF) estimates as observed in the data

p_calib Posterior samples of calibrated CSMF estimates

pcalib_postsumm Posterior summaries (mean and 95% credible interval) of calibrated CSMF estimates

va_deaths_uncalib Uncalibrated cause-specific death counts as observed in the data

va_deaths_calib_algo Algorithm-specific calibrated cause-specific death counts

va_deaths_calib_ensemble Ensemble calibrated cause-specific death counts

donotcalib A logical indicator of causes that are not calibrated for each algorithm

causes_notcalibrated Causes that are not calibrated for each algorithm

Examples

```
######### VA input as specific causes #########
# output from codEAVA() function in the EAVA package and crossVA() function in openVA package

# COMSA-Mozambique: Example (Publicly Available Version)
# Individual-Level Specific (High-Resolution) Cause of Death Data
data(comsamoz_public_openVAout)
head(comsamoz_public_openVAout$data)  # head of the data
comsamoz_public_openVAout$data[1,]  # ID and specific cause of death for individual 1

# VA-calibration for the "neonate" age group and InSilicoVA algorithm
calib_out_specific = vacalibration(va_data =
                                          setNames(list(comsamoz_public_openVAout$data),
                                               list(comsamoz_public_openVAout$va_algo)),
                                     age_group = comsamoz_public_openVAout$age_group,
                                     country = "Mozambique")

### comparing uncalibrated CSMF estimates and posterior summary of calibrated CSMF estimates
calib_out_specific$p_uncalib # uncalibrated
calib_out_specific$pcalib_postsumm["insilicova",,]

######### VA input as broad causes (output from cause_map()) #########

# COMSA-Mozambique: Example (Publicly Available Version)
# Individual-Level Broad Cause of Death Data
data(comsamoz_public_broad)
head(comsamoz_public_broad$data)
comsamoz_public_broad$data[1,]  # binary vector indicating cause of death for individual 1

# VA-calibration for the "neonate" age group and InSilicoVA algorithm
calib_out_broad = vacalibration(va_data = setNames(list(comsamoz_public_broad$data),
                                                   list(comsamoz_public_broad$va_algo)),
                                  age_group = comsamoz_public_broad$age_group,
                                  country = "Mozambique")

### comparing uncalibrated CSMF estimates and posterior summary of calibrated CSMF estimates
calib_out_broad$p_uncalib # uncalibrated
calib_out_broad$pcalib_postsumm["insilicova",,]

######### VA input as national death counts for different broad causes #########
calib_out_asdeathcount = vacalibration(va_data =
                                     setNames(list(colSums(comsamoz_public_broad$data)),
                                                   list(comsamoz_public_broad$va_algo)),
                                         age_group = comsamoz_public_broad$age_group,
                                         country = "Mozambique")

### comparing uncalibrated CSMF estimates and posterior summary of calibrated CSMF estimates
calib_out_asdeathcount$p_uncalib # uncalibrated
calib_out_asdeathcount$pcalib_postsumm["insilicova",,]
```

```
######### Example of data based on EAVA and InSilicoVA for neonates in Mozambique #########
## example VA national death count data from EAVA and InSilicoVA
va_data_example = list("eava" = c("congenital_malformation" = 40, "pneumonia" = 175,
                                  "sepsis_meningitis_inf" = 265, "ipre" = 220,
                                  "other" = 30, "prematurity" = 170),
                       "insilicova" = c("congenital_malformation" = 5, "pneumonia" = 145,
                                        "sepsis_meningitis_inf" = 370, "ipre" = 330,
                                        "other" = 60, "prematurity" = 290))

## algorithm-specific and ensemble calibration of EAVA and InSilicoVA
calib_out_ensemble = vacalibration(va_data = va_data_example,
                                   age_group = "neonate", country = "Mozambique")

### comparing uncalibrated CSMF estimates and posterior summary of calibrated CSMF estimates
calib_out_ensemble$p_uncalib # uncalibrated
calib_out_ensemble$pcalib_postsumm["eava",,] # EAVA-specific calibration
calib_out_ensemble$pcalib_postsumm["insilicova",,] # InSilicoVA-specific calibration
calib_out_ensemble$pcalib_postsumm["ensemble",,] # Ensemble calibration
```

Index