Package: vacalibration (via r-universe)

October 8, 2026

Title Calibration of Computer-Coded Verbal Autopsy Algorithm

Version 2.2

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

Description Calibrates population-level cause-specific mortality fractions (CSMFs) that are derived using computer-coded verbal autopsy (CCVA) algorithms. Leveraging the data collected in the Child Health and Mortality Prevention Surveillance (CHAMPS;<https://champshealth.org/>) project, the package stores misclassification matrix estimates of three CCVA algorithms (EAVA, InSilicoVA, and InterVA) and two age groups (neonates aged 0-27 days, and children aged 1-59 months) across countries (specific estimates for Bangladesh, Ethiopia, Kenya, Mali, Mozambique, Sierra Leone, and South Africa, and a combined estimate for all other countries), enabling global calibration. These estimates are obtained using the framework proposed in Pramanik et al. (2025;<doi:10.1214/24-AOAS2006>) and are analyzed in Pramanik et al. (2026;<doi:10.1136/bmjgh-2025-021747>). Given VA-only data for an age group, CCVA algorithm, and country, the package utilizes the corresponding misclassification matrix estimate in the modular VA-Calibration framework (Pramanik et al.,2025;<doi:10.1214/24-AOAS2006>) and produces calibrated estimates of CSMFs. The package also supports ensemble calibration to accommodate multiple algorithms. More generally, this allows calibration of population-level prevalence derived from single-class predictions of discrete classifiers. For this, users need to provide fixed or uncertainty-quantified misclassification matrices. This work is supported by the Eunice Kennedy Shriver National Institute of Child Health K99 NIH Pathway to Independence Award (1K99HD114884-01A1), the Bill and Melinda Gates Foundation (INV-034842), and the Johns Hopkins Data Science and AI Institute.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.3

Imports rstan, openVA, parallel, ggplot2, patchwork, reshape2, LaplacesDemon, MASS

Config/testthat/edition 3

Config/Needs/compile yes

Depends R (>= 3.5)

LazyData true

Suggests knitr, rmarkdown,

VignetteBuilder knitr

URL https://github.com/sandy-pramanik/vacalibration

BugReports https://github.com/sandy-pramanik/vacalibration/issues

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 https://cran.r-universe.dev

Date/Publication 2026-03-20 13:38:12 UTC

RemoteUrl https://github.com/cran/vacalibration

RemoteRef HEAD

RemoteSha e9cdff7063f1ca900ae454e16c47e8d75d83ef12

Contents

cause_map 3
CCVA_missmat 3
comsamoz_CCVAoutput 5
modular_vacalib_fixed 7
modular_vacalib_prior 8
plot_vacalib 9
plot_vacalib_fixed 11
plot_vacalib_prior 12
smart_round 12
vacalibration 13

Index **20**

cause_map *Deriving Broad Cause of Death from CCVA Outputs*

Description

Takes individual-level cause of deaths (output from CCVA algorithms) as input, and maps them to pre-defined broad causes.

Usage

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

Arguments

df	Outputs from codEAVA() in EAVA for EAVA, and codeVA() and prepCalibration() in openVA for InSilicoVA and InterVA
age_group	Character. Indicates age group. "neonate" for deaths with 0-27 days of birth, and "child" for 1-59 months of birth.

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

```
## Publicly Available Cause-of-Death (COD) Data from COMSA-Mozambique
comsamoz_CCVAoutput$neonate$eava # output from EAVA algorithm for age group "neonate"
head(comsamoz_CCVAoutput$neonate$eava)  # specific COD for the first 6 deaths

## broad cause mapping
mapped_broad_cause = cause_map(df = comsamoz_CCVAoutput$neonate$eava, age_group = "neonate")
head(mapped_broad_cause)  # broad COD for the first 6 deaths
```

CCVA_missmat *CCVA Misclassification Matrix Inventory*

Description

This is the inventory of misclassification matrix estimates for EAVA, InSilicoVA, and InterVA (doi:10.3402/gha.v5i0.19281) algorithms. The estimates are derived using the misclassification matrix modeling framework from Pramanik et al. (2025). and paired CHAMPS–VA cause-of-death data from the Child Health and Mortality Prevention Surveillance (CHAMPS) project. Please refer to Pramanik et al. (2026; doi:10.1136/bmjgh2025021747) for details on analysis. The package interpret CHAMPS and VA causes as true and estimated causes.

Usage

```
CCVA_missmat
```

Format

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 Seven specific countries: "Bangladesh", "Ethiopia", "Kenya", "Mali", "Mozambique", "Sierra Leone", and "South Africa". For all other countries, use "other".

version Character. Date stamp (yyyymmdd) for version control Only for package maintainers.

Details

Format: CCVA_missmat[[age_group]][[va_algo]][[estimate types]][[country]].

CCVA_missmat[[age_group]][[va_algo]][["postsumm"]][[country]] contains posterior summaries of misclassification matrices for a given age_group, va_algo, and country. It is an array arranged as the number of posterior summaries × CHAMPS cause × VA cause.

Neonatal causes include "congenital_malformation", "pneumonia", "sepsis_meningitis_inf", "ipre", "other", and "prematurity".

Child causes encompass "malaria", "pneumonia", "diarrhea", "severe_malnutrition", "hiv", "injury", "other", "other_infections", and "nn_causes".

For example, for "neonate" age group, "eava" algorithm in "Mozambique",

- CCVA_missmat$neonate$eava$postsumm$Mozambique[,"pneumonia","pneumonia"] are posterior summaries of the sensitivity for "pneumonia".
- CCVA_missmat$neonate$eava$postsumm$Mozambique[,"pneumonia","ipre"] are posterior summaries of the false negative rate for CHAMPS cause "pneumonia" and VA cause "ipre".

CCVA_missmat[[age_group]][[va_algo]][["postmean"]][[country]] contains posterior means of misclassification matrices for a given age_group, va_algo, and country. It is a matrix arranged as CHAMPS cause × VA cause.

For example, for "neonate" age group, "eava" algorithm in "Mozambique",

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

CCVA_missmat[[age_group]][[va_algo]][["asDirich"]][[country]] contains Dirichlet approximations of misclassification matrices for a given age_group, va_algo, and country. It is a matrix arranged as CHAMPS cause × VA cause. Each row contains Dirichlet scale parameters

that best approximates the marginal posterior of misclassification for each CHAMPS cause (rows), `age_group`, `va_algo`, and `country`.

For example, for `"neonate"` age group, `"eava"` algorithm in `"Mozambique"`, the Dirichlet distribution with scale parameters `CCVA_missmat$neonate$eava$asDirich$Mozambique["pneumonia",]` best approximates the marginal posterior of misclassification rates for CHAMPS cause `"pneumonia"`.

Specific estimates are available for seven countries: `"Bangladesh"`, `"Ethiopia"`, `"Kenya"`, `"Mali"`, `"Mozambique"`, `"Sierra Leone"`, and `"South Africa"`. For all other countries, the package uses the estimate for `"other"`. This estimate is centered at the misclassification matrix pooled across countries, and its uncertainty reflects the degree of cross-country heterogeneity observed across the seven CHAMPS countries.

Due to file size limit, the posterior samples corresponding to this inventory are available at CCVA-Misclassification-Matrices GitHub repository.

For example, `CCVA_missmat$neonate$eava$postsamples$Mozambique` contains misclassification matrix samples for `eava` among `neonate` in `Mozambique`.

The .rda file is available under the release.

References

Pramanik, S, et al. (2026) Country-Specific Estimates of Misclassification Rates of Computer-Coded Verbal Autopsy Algorithms BMJ Global Health doi:10.1136/bmjgh2025021747

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 Link

Wilson E, et al. (2025) EAVA: Deterministic Verbal Autopsy Coding with Expert Algorithm Verbal Autopsy Link

Zehang Richard Li, et al. (2024) openVA: Automated Method for Verbal Autopsy R package version 1.1.2. Link

Zehang Richard Li, et al. (2023) The openVA Toolkit for Verbal Autopsies The R Journal Link

Kalter, H., et al. (2016) Validating hierarchical verbal autopsy expert algorithms in a large data set with known causes of death. J Glob Health Link

McCormick, Tyler H., et al. (2016) Probabilistic Cause-of-Death Assignment Using Verbal Autopsies Journal of the American Statistical Association Link

Byass, Peter, et al. (2012) Strengthening standardised interpretation of verbal autopsy data: the new InterVA-4 tool Global Health Action doi:10.3402/gha.v5i0.19281

`comsamoz_CCVAoutput` *CCVA Outputs for Publicly Available Verbal Autopsy (VA) Data from COMSA–Mozambique*

Description

This contains outputs of CCVA algorithms EAVA, InSilicoVA, and InterVA (doi:10.3402/gha.v5i0.19281) when applied to publicly available verbal autopsy (VA) data collected in the Countrywide Mortality Surveillance for Action project in Mozambique (COMSA-Mozambique).

Usage

```
comsamoz_CCVAoutput
```

Format

List.

neonate List. Outputs of EAVA, InSilicoVA, and InterVA for `"neonate"` (0-27 days)

child List. Outputs of EAVA, InSilicoVA, and InterVA for `"child"` (1-59 months)

version Character. Date stamp (yyyymmdd) for version control. Only for package maintainers.

Details

Outputs for EAVA are obtained using the EAVA package, while outputs for InSilicoVA and InterVA are produced using the openVA package.

For example, `comsamoz_CCVAoutput$neonate$eava` contains output from the EAVA algorithm for "neonate".

References

Pramanik, S, et al. (2026) Country-Specific Estimates of Misclassification Rates of Computer-Coded Verbal Autopsy Algorithms BMJ Global Health doi:10.1136/bmjgh2025021747

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 Link

Wilson E, et al. (2025) EAVA: Deterministic Verbal Autopsy Coding with Expert Algorithm Verbal Autopsy Link

Zehang Richard Li, et al. (2024) openVA: Automated Method for Verbal Autopsy R package version 1.1.2. Link

Countrywide Mortality Surveillance for Action in Mozambique (COMSA-Mozambique). Link

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 doi:10.4269/ajtmh.220367

Zehang Richard Li, et al. (2023) The openVA Toolkit for Verbal Autopsies The R Journal Link

Kalter, H., et al. (2016) Validating hierarchical verbal autopsy expert algorithms in a large data set with known causes of death. Journal of Glob Health Link

McCormick, Tyler H., et al. (2016) Probabilistic Cause-of-Death Assignment Using Verbal Autopsies Journal of the American Statistical Association Link

Byass, Peter, et al. (2012) Strengthening standardised interpretation of verbal autopsy data: the new InterVA-4 tool Global Health Action doi:10.3402/gha.v5i0.19281

`modular_vacalib_fixed` *Modular VA-Calibration using Fixed Misclassification Matrix*

Description

This is a utility function. Please use vacalibration.

Usage

```
modular_vacalib_fixed(
  va_unlabeled,
  Mmat_calib,
  studycause_map,
  donotcalib,
  donotcalib_type,
  nocalib.threshold,
  path_correction,
  ensemble,
  pshrink_strength,
  nMCMC,
  nBurn,
  nThin,
  nChain,
  nCore,
  adapt_delta_stan,
  refresh_stan,
  seed,
  verbose,
  input_vacalib
)
```

Arguments

`va_unlabeled`	Same as `va_unlabeled` in `vacalibration()`
`Mmat_calib`	Same as `missmat` in `vacalibration()`
`studycause_map`	Same as `studycause_map` in `vacalibration()`
`donotcalib`	Same as `donotcalib` in `vacalibration()`
`donotcalib_type`	Same as `donotcalib_type` in `vacalibration()`
`nocalib.threshold`	Same as `nocalib.threshold` in `vacalibration()`
`path_correction`	Same as `path_correction` in `vacalibration()`
`ensemble`	Same as `ensemble` in `vacalibration()`

pshrink_strength
: Same as pshrink_strength in vacalibration()

nMCMC, nBurn, nThin
: Same as nMCMC, nBurn, and nThin in vacalibration()

nChain
: Same as nChain in vacalibration()

nCore
: Same as nCore in vacalibration()

adapt_delta_stan
: Same as adapt_delta_stan in vacalibration()

refresh_stan
: Same as refresh_stan in vacalibration()

seed
: Same as seed in vacalibration()

verbose
: Same as verbose in vacalibration()

input_vacalib
: List of inputs in vacalibration()

Value

Similar to the list returned in vacalibration()

---

modular_vacalib_prior *Modular VA-Calibration using Dirichlet Prior on Misclassification Matrix*

Description

This is a utility function. Please use vacalibration.

Usage

```
modular_vacalib_prior(
  va_unlabeled,
  Mmat_calib,
  studycause_map,
  donotcalib,
  donotcalib_type,
  nocalib.threshold,
  path_correction,
  ensemble,
  pshrink_strength,
  nMCMC,
  nBurn,
  nThin,
  nChain,
  nCore,
  adapt_delta_stan,
  refresh_stan,
  seed,
  verbose,
  input_vacalib
)
```

Arguments

va_unlabeled	Same as va_unlabeled in vacalibration()
Mmat_calib	Same as missmat in vacalibration()
studycause_map	Same as studycause_map in vacalibration()
donotcalib	Same as donotcalib in vacalibration()
donotcalib_type	Same as donotcalib_type in vacalibration()
nocalib.threshold	Same as nocalib.threshold in vacalibration()
path_correction	Same as path_correction in vacalibration()
ensemble	Same as ensemble in vacalibration()
pshrink_strength	Same as pshrink_strength in vacalibration()
nMCMC, nBurn, nThin	Same as nMCMC, nBurn, and nThin in vacalibration()
nChain	Same as nChain in vacalibration()
nCore	Same as nCore in vacalibration()
adapt_delta_stan	Same as adapt_delta_stan in vacalibration()
refresh_stan	Same as refresh_stan in vacalibration()
seed	Same as seed in vacalibration()
verbose	Same as verbose in vacalibration()
input_vacalib	List of inputs in vacalibration()

Value

Similar to the list returned in vacalibration()

---

plot_vacalib *Summary Plots of VA-Calibration*

Description

Given a VA-Calibration fit using vacalibration, this function plots misclassification matrix used in VA-Calibration, and compares uncalibrated and calibrated estimates of cause-specific mortality fractions (CSMFs).

Usage

```
plot_vacalib(vacalib_fit, toplot = "both")
```

Arguments

vacalib_fit	Fitted object from vacalibration()
toplot	Character. What to plot. When toplot="missmat" and missmat_type="fixed", it plots the fixed misclassification matrix used in calibration. When missmat_type equals "fixed" or "samples", it plots the average misclassification matrix. When toplot="csmf", it compares uncalibrated and calibrated estimates of cause-specific mortality fractions (CSMFs). When "both", it plots both the misclassification matrix and estimates of CSMFs.

Value

It returns a plot comparing misclassification matrix used in calibration, and uncalibrated and calibrated estimates of cause-specific mortality fractions (CSMFs).

Examples

```
######### COMSA-Mozambique VA-COD data #########
data(comsamoz_CCVAoutput)

######### Algorithm-Specific Calibration #########

# EAVA
vacalib_out_eava = vacalibration(va_data = comsamoz_CCVAoutput$neonate[1],
                                 age_group = "neonate", country = "Mozambique",
                                 saveoutput = FALSE)
print(vacalib_out_eava$input$missmat_type)
print(vacalib_out_eava$input)
print(names(vacalib_out_eava$input))

# summary plot
plot_vacalib(vacalib_fit = vacalib_out_eava, toplot = "missmat")  # misclassification matrix
plot_vacalib(vacalib_fit = vacalib_out_eava, toplot = "csmf")  # CSMFs
plot_vacalib(vacalib_fit = vacalib_out_eava, toplot = "both")  # both


# InSilicoVA
vacalib_out_insilicova = vacalibration(va_data = comsamoz_CCVAoutput$neonate[2],
                                       age_group = "neonate", country = "Mozambique",
                                       saveoutput = FALSE)

# summary plot
plot_vacalib(vacalib_fit = vacalib_out_insilicova, toplot = "missmat")  # misclassification matrix
plot_vacalib(vacalib_fit = vacalib_out_insilicova, toplot = "csmf")  # CSMFs
plot_vacalib(vacalib_fit = vacalib_out_insilicova, toplot = "both")  # both


# InterVA
vacalib_out_interva = vacalibration(va_data = comsamoz_CCVAoutput$neonate[3],
```

```
                                    age_group = "neonate", country = "Mozambique",
                                    saveoutput = FALSE)

# summary plot
plot_vacalib(vacalib_fit = vacalib_out_interva, toplot = "missmat")  # misclassification matrix
plot_vacalib(vacalib_fit = vacalib_out_interva, toplot = "csmf")  # CSMFs
plot_vacalib(vacalib_fit = vacalib_out_interva, toplot = "both")  # both



######### Ensemble Calibration #########
vacalib_out_ensemble = vacalibration(va_data = comsamoz_CCVAoutput$neonate,
                                     age_group = "neonate", country = "Mozambique",
                                     saveoutput = FALSE)

# summary plot
plot_vacalib(vacalib_fit = vacalib_out_ensemble, toplot = "missmat")  # misclassification matrix
plot_vacalib(vacalib_fit = vacalib_out_ensemble, toplot = "csmf")  # CSMFs
plot_vacalib(vacalib_fit = vacalib_out_ensemble, toplot = "both")  # both
```

`plot_vacalib_fixed` *Summary Plots of VA-Calibration Using Fixed Misclassification Matrix*

Description

This is a utility function. Please use plot_vacalib.

Usage

```
plot_vacalib_fixed(vacalib_fit, toplot)
```

Arguments

`vacalib_fit`	Fitted object from `vacalibration()`
`toplot`	Character. Same as `toplot` in `plot_vacalib_fixed()`

Value

Plots misclassification matrices and/or cause-specific mortality fractions

plot_vacalib_prior *Summary Plots of VA-Calibration Using Dirichlet Prior on Misclassification Matrix*

Description

This is a utility function. Please use plot_vacalib.

Usage

```
plot_vacalib_prior(vacalib_fit, toplot)
```

Arguments

vacalib_fit	Fitted object from vacalibration()
toplot	Character. Same as toplot in plot_vacalib_fixed()

Value

Plots misclassification matrices and/or cause-specific mortality fractions

smart_round *Round and maintain a target sum*

Description

Rounds a vector to the specified number of decimal places and maintains the sum it had before rounding.

Usage

```
smart_round(x, target_sum, digits = 0)
```

Arguments

x	Numeric vector.
target_sum	Numeric. The target sum to be maintained after rounding. Default is NULL which sets target_sum=sum(x).
digits	Positive integer. Indicates the number of decimal places to be used.

Value

Numeric vector.

Examples

```
x = rep(1/3, 3)
round(x, 2)
smart_round(x, 1, 2)
```

`vacalibration` *VA-Calibration*

Description

This is the main function in the package. It calibrates population-level cause-specific mortality fractions (CSMFs) that are derived using computer-coded verbal autopsy (CCVA) algorithms. For VA-Calibration, the function utilizes the inventory of misclassification matrix estimates CCVA_missmat. The outputs from EAVA and openVA for InSilicoVA and InterVA can be input directly (see below). This seamlessly supports VA-Calibration for EAVA, InSilicoVA, and InterVA (doi:10.3402/gha.v5i0.19281). For other CCVA algorithms, the input expects either an individual by cause matrix, or cause-specific death count vector (see below). When broad-cause-specific death counts are input and they do not match the broad causes in the stored misclassification estimates, then either `studycause_map` or the misclassification matrices (fixed or as row-specific Dirichlet priors) need to be provided. More generally, this allows us to calibrate population-level prevalence derived from single-class predictions of discrete classifiers. For this, users need to provide fixed or uncertainty-quantified misclassification matrices.

Usage

```
vacalibration(
  va_data = NULL,
  age_group = NULL,
  country = NULL,
  missmat_type = c("prior", "fixed", "samples")[1],
  studycause_map = NULL,
  missmat = NULL,
  donotcalib = NULL,
  donotcalib_type = c("learn", "fixed")[1],
  nocalib.threshold = 0.1,
  path_correction = TRUE,
  ensemble = NULL,
  pshrink_strength = NULL,
  nMCMC = 5000,
  nBurn = 5000,
  nThin = 1,
  nChain = 1,
  nCore = 1,
  adapt_delta_stan = 0.9,
  refresh_stan = NULL,
  seed = 1,
```

```
  verbose = TRUE,
  saveoutput = FALSE,
  output_filename = NULL,
  output_dir = NULL
)
```

Arguments

va_data	Named list. Algorithm-specific unlabeled VA data. It expects a named list, such as `list("algo1" = algo1_output, "algo2" = algo2_output, ...)`. Misclassification matrix estimates in `CCVA_missmat` are only available for CCVA algorithms EAVA, InSilicoVA, and InterVA (doi:10.3402/gha.v5i0.19281). For them the algorithm names in input data must be `"eava"`, `"insilicova"`, and `"interva"`. Otherwise, users must input misclassification matrices in `missmat` (see more details in `missmat`). VA data provided for each algorithm (`algo1_output`, `algo2_output`, ...) can be either 1. outputs of CCVA algorithms (output from `codEAVA()` in `EAVA` for EAVA, and `codeVA()` and `prepCalibration()` in `openVA` for InSilicoVA and InterVA), or 2. individual broad cause of deaths (output from `cause_map`), or 3. a vector of cause-specific death counts. More generally, it can calibrate for any discrete classifier. In that case, the input must be one of these two types: 1. A binary matrix arranged as individuals along rows and class labels as columns. For each individual (row), 1 occurs exactly once and it indicates the estimated class label. Other elements in the row are 0. 2. A vector of label-specific counts. This indicates the estimated number of individuals for each label.
age_group	Character. When `missmat` is `NULL`, this indicates the age group for which the misclassification matrix estimates in `"CCVA_missmat"` should be applied (default). It can be either `"neonate"` for neonatal deaths occurring between 0-27 days after birth, and `"child"` for deaths among children occurring between 1-59 months.
country	Character. When `missmat` is `NULL`, this indicates the country for which the misclassification matrix estimates in `"CCVA_missmat"` should be applied (default). If input is "Bangladesh", "Ethiopia", "Kenya", "Mali", "Mozambique", "Sierra Leone", or "South Africa", then their corresponding misclassification matrix is applied. For any other country, the estimate for "other" is applied (see `"CCVA_missmat"` for more details).
missmat_type	Character. Indicates the type of misclassification matrix estimates provided in `missmat`. `"prior"` (default) Dirichlet priors for each row of the misclassification matrix.

"fixed" A fixed misclassification matrix.

"samples" Random samples of misclassification matrix.

Uncertainty in misclassification matrix estimates is only propagated for "prior" or "samples".

studycause_map Named character vector. A mapping of observed causes (in va_data) to broad causes (for which misclassification estimates are available in "CCVA_missmat").

Required only when missmat is NULL, and causes observed in va_data are not a subset of broad causes in "CCVA_missmat" (see "CCVA_missmat" for list of causes).

For example, if causes observed in va_data for neonates are "cause1", "cause2", "cause3", and "cause4", studycause_map expects input as c("cause1" = "pneumonia", "cause2" = "ipre", "cause3" = "other", "cause4" = "other").

missmat Named list. Similarly structured as va_data. For example, list("algo1" = missmat_algo1, "algo2" = missmat_algo2, ...).

For missmat_type = "prior", missmat_algo1, missmat_algo2, ... are matrices with positive entries and arranged as CHAMPS cause × VA cause. Each row of the matrix is a vector of Dirichlet scale parameters. This the Dirichlet prior assumed on the corresponding row of the misclassification matrix. See stored estimates CCVA_missmat$neonate$eava$asDirich$Mozambique for example.

For missmat_type = "fixed", missmat_algo1, missmat_algo2, ... are misclassification matrices arranged as CHAMPS cause × VA cause. See stored estimates CCVA_missmat$neonate$eava$postmean$Mozambique for example.

For missmat_type = "samples", missmat_algo1, missmat_algo2, ... are arrays of misclassification matrix samples arranged as samples × CHAMPS cause × VA cause. missmat_algo1[i,,] is the i-th sample of misclassification matrix for algo1. See the samples stored in the CCVA-Misclassification-Matrices GitHub repository for example.

Names and length of missmat must be identical to va_data.

Users are not required to provide missmat for using the stored estimates in "CCVA_missmat". They can simply input the required age_group, country, missmat_type, and studycause_map accordingly.

missmat needs to be input when causes observed in va_data are not a subset of CHAMPS broad causes (in "CCVA_missmat") and studycause_map is not provided.

For a general purpose of calibrating categorical classifiers, CHAMPS and VA causes can be interpreted as true and estimated labels and users must input missmat.

donotcalib Named list. List of causes for each algorithm that users do not want to calibrate. The set of causes can differ across algorithms.

Default: list("eava"="other", "insilicova"="other", "interva"="other"). When using the stored estimates in CCVA_missmat, this implies that the cause-specific mortality fractions (CSMF) for CHAMPS broad cause "other" is not calibrated.

When causes observed in va_data are not a subset of CHAMPS broad causes and studycause_map is provided, all observed causes in va_data that match with the causes in donotcalib are not calibrated.

Set `list("eava"=NULL, "insilicova"=NULL, "interva"=NULL)` to calibrate all causes.

For a general purpose of calibrating categorical classifiers, causes can be interpreted as class labels and specified accordingly.

`donotcalib_type`

Character. `"learn"` (default) or `"fixed"`.

For `donotcalib_type="fixed"`, only the causes specified in `"donotcalib"` are not calibrated.

For `donotcalib_type="learn"`, it learns additional causes from misclassification matrix in `"missmat"` that cannot be calibrated.

When misclassification rates for a VA cause do not change across CHAMPS causes, the calibration equation becomes underdetermined (see the footnote on pg. 1227 in Pramanik et al. (2025)). When `donotcalib_type="learn"`, it screens VA causes that do not vary beyond `nocalib.threshold`. These causes are added to the `donotcalib` list.

For a general purpose of calibrating categorical classifiers, causes can be interpreted as class labels and specified accordingly.

`nocalib.threshold`

Numeric in `(0,1)`.

The threshold used to screen VA causes when `donotcalib_type="learn"`.

Default: 0.1.

`path_correction`

Logical. Setting `TRUE` shrinks misclassification matrix towards the identity matrix to improve stability in VA-Calibration.

Default is `TRUE`.

`ensemble` Logical. Whether to perform ensemble calibration when outputs from multiple algorithms are provided.

Default is `TRUE`.

`pshrink_strength`

Positive numeric. Degree of shrinkage of calibrated CSMF estimates towards its uncalibrated estimates. This is the parameter `eta` in the prior of calibrated CSMF `p` (see pg. 1226 in Pramanik et al. (2025)).

Only used when `path_correction=FALSE`. `pshrink_strength` is set to 0 when `path_correction=TRUE`.

Defaults to 4 when `path_correction=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.

`nChain` Positive integer. Number of chains for Stan sampling. Default 1.

`nCore` Positive integer. Number of cores to run multiple chains in parallel for Stan sampling. Default 1.

`adapt_delta_stan`
: Numeric in `(0,1)`. `adapt_delta` parameter in `rstan`.
Influences the behavior of the No-U-Turn Sampler (NUTS) in Stan.
Default 0.9.

`refresh_stan`
: Positive integer. Print every `refresh_stan`% progress.
Default 20.

`seed`
: Numeric. `seed` parameter in rstan. Default 1.

`verbose`
: Logical. Whether to report progress (`TRUE`) or not (`FALSE`).
Default `TRUE`.

`saveoutput`
: Logical. Save output (`TRUE`) or not (`FALSE`).
Default `TRUE`.

`output_filename`
: Character. Output name to save as.
Default `vacalibration_out`.

`output_dir`
: Output directory or file path to save at.
Default `getwd()`, the working directory.

Value

A list with components:

- `calib_MCMCout` — Output from Stan fits.
- `p_uncalib` — Uncalibrated estimates of CSMF. It is a matrix arranged as algorithm × VA causes (estimated labels).
- `p_calib` — Posterior samples of calibrated CSMF. It is an array arranged as algorithm × samples × VA causes (or estimated labels).
- `pcalib_postsumm` — Posterior summaries (mean and 95% credible interval) of calibrated CSMF. It is an array arranged as algorithm × summary measures × VA causes (or estimated labels).
- `va_deaths_uncalib` — Uncalibrated cause-specific death counts. It is a matrix arranged as algorithm × VA causes (or estimated labels).
- `va_deaths_calib_algo` — Calibrated cause-specific death counts from algorithm-specific calibration. It is a matrix arranged as algorithm × VA causes (or estimated labels).
- `va_deaths_calib_ensemble` — Calibrated cause-specific death counts from ensemble calibration. It is a matrix arranged as algorithm × VA causes (or estimated labels).
- `Mmat_input` — `"missmat"` as provided in the input. It is an array arranged as algorithm × CHAMPS cause (or true labels) × VA causes (or estimated labels).
- `Mmat_study` — Modified `Mmat_input` if `studycause_map` is provided. It is an array arranged in the same way as `Mmat_input`.
- `Mmat_tomodel` — Modified `Mmat_study` if `path_correction` is `TRUE`. This is used for calibration. It is an array arranged in the same way as `Mmat_input` and `Mmat_study`.
- `donotcalib_study` — This indicates causes that are not calibrated for each algorithm, as specified in the input `donotcalib`. It is a logical matrix arranged as algorithm × VA causes (or estimated labels).

- `donotcalib_tomodel` — This indicates causes that are not calibrated in each calibration. This is a modified `donotcalib_study` if `donotcalib_type` is provided and `ensemble=TRUE`. It is a logical matrix arranged as algorithm × VA causes (or estimated labels).
- `calibrated` — `TRUE` or `FALSE` indicating whether Stan sampling was performed for calibration.
- `lambda_calibpath` — When `path_correction=TRUE`, this indicates the degree of shrinkage of CSMF for each algorithm towards uncalibrated estimates. This is a vector of numerics in `[0,1]` showing degrees of shrinkage for each algorithm.
- `K` — Number of algorithms.
- `nCause` — Number of causes.
- `causes` — Name of causes.
- `input` — List of inputs.

References

Pramanik, S, et al. (2026) Country-Specific Estimates of Misclassification Rates of Computer-Coded Verbal Autopsy Algorithms BMJ Global Health doi:10.1136/bmjgh2025021747

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 Link

Fiksel, J., et al. (2022) Generalized Bayes Quantification Learning under Dataset Shift Journal of the American Statistical Association Link

Datta, A, et al. (2021) Regularized Bayesian transfer learning for population-level etiological distributions. Biostatistics doi:10.1093/biostatistics/kxaa001

Examples

```
######### COMSA-Mozambique VA-COD data #########
data(comsamoz_CCVAoutput)

# neonatal deaths
comsamoz_CCVAoutput$neonate$eava  # output from running EAVA
comsamoz_CCVAoutput$neonate$insilicova  # output from running InSilicoVA
comsamoz_CCVAoutput$neonate$interva  # output from running InterVA



######### Algorithm-Specific Calibration #########

# EAVA
vacalib_out_eava = vacalibration(va_data = comsamoz_CCVAoutput$neonate[1],
                                 age_group = "neonate", country = "Mozambique",
                                 saveoutput = FALSE)

## CSMF
vacalib_out_eava$p_uncalib   # uncalibrated
vacalib_out_eava$p_calib   # calibrated
vacalib_out_eava$pcalib_postsumm   # summary of calibrated estimates
```

```
## death counts
vacalib_out_eava$va_deaths_uncalib   # uncalibrated
vacalib_out_eava$va_deaths_calib_algo   # calibrated


# InSilicoVA
vacalib_out_insilicova = vacalibration(va_data = comsamoz_CCVAoutput$neonate[2],
                                       age_group = "neonate", country = "Mozambique",
                                       saveoutput = FALSE)

## CSMF
vacalib_out_insilicova$p_uncalib   # uncalibrated
vacalib_out_insilicova$p_calib   # calibrated
vacalib_out_insilicova$pcalib_postsumm   # summary of calibrated estimates

## death counts
vacalib_out_insilicova$va_deaths_uncalib   # uncalibrated
vacalib_out_insilicova$va_deaths_calib_algo   # calibrated


# InterVA
vacalib_out_interva = vacalibration(va_data = comsamoz_CCVAoutput$neonate[3],
                                    age_group = "neonate", country = "Mozambique",
                                    saveoutput = FALSE)

## CSMF
vacalib_out_interva$p_uncalib   # uncalibrated
vacalib_out_interva$p_calib   # calibrated
vacalib_out_interva$pcalib_postsumm   # summary of calibrated estimates

## death counts
vacalib_out_interva$va_deaths_uncalib   # uncalibrated
vacalib_out_interva$va_deaths_calib_algo   # calibrated



######### Ensemble Calibration #########
vacalib_out_ensemble = vacalibration(va_data = comsamoz_CCVAoutput$neonate,
                                     age_group = "neonate", country = "Mozambique",
                                     saveoutput = FALSE)

## CSMF
vacalib_out_ensemble$p_uncalib   # uncalibrated
vacalib_out_ensemble$p_calib   # calibrated
vacalib_out_ensemble$pcalib_postsumm   # summary of calibrated estimates

## death counts
vacalib_out_ensemble$va_deaths_uncalib   # uncalibrated
vacalib_out_ensemble$va_deaths_calib_algo   # algorithm-specific calibrated death counts
vacalib_out_ensemble$va_deaths_calib_ensemble   # ensemble calibrated death counts
```

Index

∗ **datasets**
CCVA_missmat, 3
comsamoz_CCVAoutput, 5

cause_map, 3
CCVA_missmat, 3, *13*
comsamoz_CCVAoutput, 5

modular_vacalib_fixed, 7
modular_vacalib_prior, 8

plot_vacalib, 9, *11*, *12*
plot_vacalib_fixed, 11
plot_vacalib_prior, 12

smart_round, 12

vacalibration, *7–9*, 13