Package 'sardine'

November 27, 2026

Title Epigenetic Clock and Phenotype Associations from EWAS Summary Statistics

Version 0.1.0

Description Estimates the association between an epigenetic clock and a phenotype using only summary statistics from an epigenome-wide association study (EWAS), the clock's CpG weights, and a reference DNA methylation covariance matrix. No individual-level data is required. Handles linear, logistic and Cox proportional hazards EWAS models, either regression direction, and beta-value or M-value methylation scales.

License GPL (>= 3)

URL https://github.com/qlu-lab/SARDINE

BugReports https://github.com/qlu-lab/SARDINE/issues

Encoding UTF-8

Depends R (>= 4.0.0)

Imports data.table, R.utils, stats, utils

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

Config/roxygen2/version 8.1.0

NeedsCompilation no

Author Ryan Moreno [aut, cre], Yuchang Wu [aut]

Maintainer Ryan Moreno <rrmoreno@wisc.edu>

Repository CRAN

Date/Publication 2026-11-27 18:20:02 UTC

Contents

print.sardine_result 2
sardine 2

Index **6**

print.sardine_result *Print a SARDINE result*

Description

Print a SARDINE result

Usage

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

Arguments

x	An object of class sardine_result.
...	Passed to print.data.frame().

Value

x, invisibly.

sardine *Summary-statistics-based association between epigenetic clock and phenotype*

Description

Estimates the association between an epigenetic clock and a phenotype using only EWAS summary statistics, the clock's CpG weights, and a reference DNA methylation covariance matrix. No individual-level data is required.

Usage

```
sardine(
  ewas_sumstats,
  ewas_direction = c("M~Y", "Y~M"),
  ewas_model = c("linear", "logistic", "cox"),
  ewas_DNAm_scale = c("beta", "m"),
  ewas_pheno_type = c("continuous", "binary"),
  clock,
  DNAm_cov_matrix,
  mean_beta = NULL,
  varY = NULL,
  scale = FALSE,
  age_adult = 20,
```

```
  N = NULL,
  cox_pi = NULL,
  cox_E = NULL,
  verbose = TRUE
)
```

Arguments

`ewas_sumstats`	EWAS summary statistics: can be a dir/to/file or a data.table, data.frame; must contain columns CpG, BETA, SE, and N.
`ewas_direction`	Direction of the association (outcome ~ exposure) in the EWAS model: `"M~Y"` (methylation regressed on phenotype; default) or `"Y~M"` (phenotype regressed on methylation).
`ewas_model`	EWAS regression model: `"linear"` (default), `"logistic"`, or `"cox"` (Cox proportional hazards).
`ewas_DNAm_scale`	Scale of the EWAS DNA methylation data: `"beta"` (beta values; default) or `"m"` (M-values).
`ewas_pheno_type`	EWAS phenotype type: `"continuous"` (default) or `"binary"`. If binary, will also output an estimate of the log-odds ratio of clock on phenotype. For all four options above, matching ignores case and whitespace (e.g. `"y ~ m"` is accepted as `"Y~M"`).
`clock`	Epigenetic clock weights: can be a dir/to/clock/file or a data.frame/data.table; must contain columns CpG, CoefficientTraining.
`DNAm_cov_matrix`	Variance-covariance matrix for DNAm. Must have the same covariate adjustments as the ewas summary stats. Can be a dir/to/file of a delimited table (e.g. TSV) whose first column holds the CpG IDs and whose header lists the same CpG IDs in the same order; a dir/to/file.rds containing a matrix; a matrix with CpG rownames and colnames; a data.frame of numeric columns with CpG rownames; or a data.frame/data.table laid out like the delimited file.
`mean_beta`	Optional. dir/to/file containing mean DNAm beta-value at each CpG site (expected to be in beta-scale). Needed for M-value to beta-value conversion. A data.frame or data.table with 2 columns: CpGmarker, beta_bar. Best if computed from the same samples as the EWAS itself. Default: `NULL`; only needed for M-value to beta-value conversion.
`varY`	Variance of the phenotype used in EWAS: raw phenotype if no covariates; residualized phenotype if adjusted for covariates in the EWAS model. Default: `NULL`; we can estimate varY using only EWAS sumstats and DNAm reference matrix.
`scale`	Logical. For Horvath or PedBE clocks, we can scale the point estimate and SE; doesn't affect Pvalue.
`age_adult`	Only used if `scale = TRUE`.
`N`	Optional. EWAS total sample size. Default is to read this from the EWAS summary statistics.

cox_pi	Only used when ewas_model = "cox". Proportion of cases at the end of the study. If not provided, this is computed as cox_E/N. Exactly one of cox_pi and cox_E must be provided if ewas_model = "cox".
cox_E	Only used when ewas_model = "cox". Number of cases at the end of the study. Default NULL. Exactly one of cox_pi and cox_E must be provided if ewas_model = "cox".
verbose	True/False. Whether to print progress and diagnostic messages.

Details

Two coefficients are always returned: lambda, from the regression of the clock on the phenotype, and gamma, from the regression of the phenotype on the clock. Binary phenotypes additionally yield gamma_logistic, a log-odds ratio. Cox-model inputs additionally yield log_HR, a clock-level log hazard ratio. All rows share the same Z statistic and p-value by construction.

See https://github.com/qlu-lab/SARDINE/blob/main/inputs.md for details on every input.

Value

An object of class sardine_result: a data.frame with one row per estimated parameter (lambda and gamma always; gamma_logistic for binary phenotypes; log_HR for Cox EWAS) and columns:

model EWAS model used, e.g. "Phenotype ~ Clock (Cox)".

param Parameter being estimated, e.g. "log_HR".

coef Point estimate for the parameter.

SE Estimated standard error for the point estimate.

z Z statistic (estimate / SE). All rows share the z of the linear gamma test; for gamma_logistic the SE comes from the delta method, so z != coef / SE for that row.

p Two-sided normal p-value for the estimate, testing the null hypothesis that the effect is 0.

neglog10p -log10(p), computed directly from z so it stays finite when very small p-values underflow to 0.

N EWAS sample size: the N argument if supplied, otherwise the median per-CpG N across the CpGs used.

note Notes about the specific settings for this row (e.g. log-odds vs log_HR), or NA.

Retrieve the metadata with attr(result, "metadata"): a one-row data.frame with columns:

sardine_version Version of SARDINE that produced the result.

N_CpGs_ewas, N_CpGs_clock, N_CpGs_DNAmRef Number of CpGs in the EWAS, clock, and DNAm reference respectively.

N_CpGs_ewas_clock, N_CpGs_ewas_ref, N_CpGs_clock_ref Number of CpGs overlapping between each pair of inputs.

N_CpGs_final Number of CpGs in all three inputs; these are the CpGs actually used for the association analysis.

ClockCpG_coverage_ratio Fraction of the clock's CpGs that ended up in the final set of CpGs used.

ClockCpG_missed_ratio 1 - `ClockCpG_coverage_ratio`.

cox_pi Proportion of cases at the end of the study, either input directly or computed as `cox_E / median(N)`. `NA` if the EWAS model is not Cox.

kappa_cox Cox conversion factor `(1 - cox_pi) * -log(1 - cox_pi)` used to convert between the linear-scale gamma and the Cox log hazard ratio. `NA` if the EWAS model is not Cox.

varY Variance of the phenotype (after any capping), either input directly or estimated from the EWAS summary statistics and the DNAm reference matrix.

varY_type `"user_provided"` or `"inferred"`.

varY_ratio (varY prior to capping) / (theoretical maximum varY), where the maximum is the varY you'd get if the clock explained all of the phenotype's variance. Values above 1 trigger capping. `NA` if the clock has no correlation with the phenotype.

varY_note Note about whether varY had to be capped at the theoretical maximum, or `NA` if no capping was needed.

Cox models

For a Cox Proportional Hazards model, `ewas_direction` must be `"Y~M"`, `ewas_model` must be `"cox"`, `ewas_pheno_type` must be `"binary"`, and `N` should represent the total sample size at the end of the study. Exactly one of `cox_pi` and `cox_E` must be provided. `cox_pi = cox_E/N` is the proportion of cases at the end of the study. It is recommended that `varY` is inferred rather than provided as user-supplied so that the `log_hr` estimate is invariant to `cox_pi`.

Logistic models

for a Logistic model, `ewas_direction` must be `"Y~M"`, `ewas_model` must be `"logistic"`, and `ewas_pheno_type` must be `"binary"`.

Examples

```
# Small synthetic inputs bundled with the package
inputs <- system.file("extdata", "fixtures", "inputs", package = "sardine")

result <- sardine(
  ewas_sumstats   = file.path(inputs, "ewas_sumstats_baseline_synthetic.tsv"),
  clock           = file.path(inputs, "clock_weights_baseline_synthetic.tsv"),
  DNAm_cov_matrix = file.path(inputs, "sigma_baseline_synthetic.tsv"),
  verbose         = FALSE
)
result
attr(result, "metadata")
```

Index

print.sardine_result, 2

sardine, 2