

Package: gibbsTI (via r-universe)

September 22, 2026

Title Calibrated Gibbs Posteriors for Tolerance Intervals

Version 0.1.0

Description Provides tools for constructing one-sided and two-sided nonparametric tolerance intervals using calibrated Gibbs posteriors. The methods target population quantiles and content levels while achieving nominal frequentist coverage through data-driven calibration. The methodology is described in Pourmohamad, Richardson, and Sansó (2026) [doi:10.1080/00401706.2026.2699103](https://doi.org/10.1080/00401706.2026.2699103).

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

LinkingTo Rcpp

Imports Rcpp

Repository <https://tpourmohamad.r-universe.dev>

Date/Publication 2026-07-13 18:53:01 UTC

RemoteUrl <https://github.com/tpourmohamad/gibbsti>

RemoteRef HEAD

RemoteSha ea0adec2318299cc003608a5cf30f270190220a8

Contents

.calibrate_eta_gpc	2
.run_sampler	2
gibbsTI	3
Index	6

<code>.calibrate_eta_gpc</code>	<i>Internal wrapper for GPC calibration</i>
---------------------------------	---

Description

Internal wrapper for GPC calibration

Usage

```
.calibrate_eta_gpc(  
  x,  
  side,  
  target,  
  content,  
  confidence,  
  tau,  
  tau_lower,  
  tau_upper,  
  verbose,  
  eta0,  
  max_iter,  
  B,  
  tol,  
  c,  
  gamma,  
  prop_sd,  
  w,  
  m,  
  n_samps_boot,  
  burn_in_boot  
)
```

<code>.run_sampler</code>	<i>Internal wrapper for final MCMC sampling</i>
---------------------------	---

Description

Internal wrapper for final MCMC sampling

Usage

```
.run_sampler(  
  x,  
  side,  
  target,
```

```

    tau,
    tau_lower,
    tau_upper,
    eta,
    n_mcmc,
    burnin,
    thin,
    w,
    m,
    prop_sd,
    verbose
)

```

gibbsTI

Fit a Gibbs Posterior Tolerance Interval

Description

Constructs a one-sided or two-sided nonparametric tolerance interval from a calibrated Gibbs posterior. The learning rate η is chosen automatically by Gibbs posterior calibration (GPC) so that the interval attains the nominal frequentist coverage, or it can be fixed by the user.

Usage

```

gibbsTI(
  x,
  side = c("one", "two"),
  type = c("upper", "lower"),
  target = c("content", "quantile"),
  content = 0.95,
  confidence = 0.9,
  tau = NULL,
  tau_lower = NULL,
  tau_upper = NULL,
  eta_method = c("GPC", "fixed"),
  eta = NULL,
  max_iter = 15,
  control = list(),
  n_mcmc = 5000,
  burnin = 1000,
  thin = 1,
  seed = NULL,
  verbose = TRUE
)

```

Arguments

<code>x</code>	Numeric vector of observations (at least 5, all finite).
<code>side</code>	Direction of the interval: "one" (one-sided) or "two" (two-sided, centered).
<code>type</code>	For one-sided intervals only, "upper" (bound above) or "lower" (bound below). Ignored when <code>side = "two"</code> .
<code>target</code>	Calibration target: "content" (default; the interval is defined by a population proportion) or "quantile" (the interval targets explicit population quantiles supplied through <code>tau</code> , <code>tau_lower</code> , <code>tau_upper</code>).
<code>content</code>	Population proportion the interval should contain, in (0, 1) (e.g. 0.95). Used when <code>target = "content"</code> ; when <code>target = "quantile"</code> it only supplies defaults for any unset <code>tau_*</code> argument.
<code>confidence</code>	Frequentist confidence level in (0, 1) (e.g. 0.90). This is both the GPC coverage target ($1 - \alpha$) and the posterior level read off as the reported limit.
<code>tau</code>	Target quantile for a one-sided "quantile" interval, in (0, 1). Only used when <code>target = "quantile"</code> ; ignored when <code>target = "content"</code> .
<code>tau_lower, tau_upper</code>	Lower/upper target quantiles for a two-sided "quantile" interval, in (0, 1) with <code>tau_lower < tau_upper</code> . Only used when <code>target = "quantile"</code> ; ignored when <code>target = "content"</code> .
<code>eta_method</code>	How the Gibbs learning rate <code>eta</code> is chosen: "GPC" (default; data-driven calibration) or "fixed" (use the supplied <code>eta</code>).
<code>eta</code>	Gibbs learning rate (positive). Optional starting value when <code>eta_method = "GPC"</code> (defaults to 1); required when <code>eta_method = "fixed"</code> .
<code>max_iter</code>	Maximum number of Robbins-Monro iterations for the GPC search (default 15). Only used when <code>eta_method = "GPC"</code> .
<code>control</code>	A list of control parameters for the MCMC samplers and GPC: • <code>B</code>: Number of bootstrap replicates for calibration (default 200). • <code>tol</code>: Convergence tolerance for <code>eta</code> (default $1e-3$). • <code>c, gamma</code>: Step-size parameters for Robbins-Monro (defaults 0.5, 0.75). • <code>w, m</code>: Slice sampler width and max steps for one-sided intervals. • <code>prop_sd</code>: Proposal standard deviation for Metropolis-Hastings in the two-sided case. • <code>n_samps_calib</code>: MCMC samples used per bootstrap during calibration (default 1000). • <code>burnin_calib</code>: Burn-in used per bootstrap during calibration (default 200).
<code>n_mcmc</code>	Number of posterior draws retained for the final interval, after burn-in and thinning (default 5000).
<code>burnin</code>	Number of initial iterations discarded as burn-in (default 1000).
<code>thin</code>	Thinning interval: every <code>thin</code> -th post-burn-in draw is kept (default 1).
<code>seed</code>	Optional integer random seed for reproducibility.
<code>verbose</code>	Logical; if TRUE, prints progress messages.

Value

An object of class "gibbsTI": a list with the fitted interval (a numeric limit for one-sided fits, or a list with lower and upper for two-sided fits), the calibrated eta, the retained posterior samples, the data, the quantile level(s) tau_used, calibration diagnostics (calibrated_coverage, eta_history), and the settings used. Use `print()`, `summary()`, and `plot()` on the result.

Examples

```
set.seed(1)
x <- rnorm(50)

# Fast: fix the learning rate (no calibration search)
fit <- gibbsTI(x, side = "one", type = "upper", eta_method = "fixed",
              eta = 1, n_mcmc = 500, burnin = 200, verbose = FALSE)
print(fit)

# Data-driven GPC calibration (slower) for a two-sided 95%/90% interval
fit2 <- gibbsTI(x, side = "two", content = 0.95, confidence = 0.90,
               seed = 1, verbose = FALSE)
summary(fit2)
```

Index

`.calibrate_eta_gpc`, [2](#)

`.run_sampler`, [2](#)

`gibbsTI`, [3](#)

`plot()`, [5](#)

`print()`, [5](#)

`summary()`, [5](#)