

# Package ‘NAPrior’

May 7, 2026

**Title** Network Meta-Analytic Predictive Prior for Mid-Trial SoC Changes

**Version** 0.2.0

**Description** Implements the Network meta-Analytic Predictive (NAP) prior framework to accommodate changes in the standard of care (SoC) during ongoing randomized controlled trials (RCTs). The method synthesizes pre- and post-change in-trial data by leveraging external evidence, particularly head-to-head trials comparing the original and new standards of care, to bridge the two evidence periods and enable principled borrowing. The package provides utilities to construct NAP-based priors and perform Bayesian inference for time-to-event endpoints using summarized trial evidence.

**License** MIT + file LICENSE

**Encoding** UTF-8

**Depends** R (>= 4.0)

**Imports** stats, graphics, survival, R2jags, metafor, purrr, dplyr, tibble

**Suggests** rjags, knitr, rmarkdown

**VignetteBuilder** knitr

**RoxygenNote** 7.3.2

**NeedsCompilation** no

**Author** Chunyi Zhang [aut, cre]

**Maintainer** Chunyi Zhang <czhang12@mdanderson.org>

**Repository** CRAN

**Date/Publication** 2026-02-14 19:30:02 UTC

## Contents

NAP_oc . . . . .	2
NAP_posterior . . . . .	3
NAP_prior . . . . .	4
post_w . . . . .	6
tune_param_eNAP . . . . .	8
<b>Index</b>	<b>10</b>

**Description**

Runs Monte Carlo simulations of an E vs C2 trial and performs Bayesian analysis with a NAP-based prior constructed by `NAP_prior()`. The routine supports both single external study setting and multiple external studies settings as encoded in the provided `NAP_prior` object, and works with either a fixed mixture weight (mNAP) or an elastic, data-adaptive weight (eNAP).

**Usage**

```
NAP_oc(
  NAP_prior = NULL,
  theta_EC2 = 0,
  n_EC2 = 200,
  lambda = 2,
  sim_model = c("Exponential", "Weibull"),
  model_param = 0.05,
  iter = 2000,
  chains = 4,
  seed = 123,
  nsim = 100,
  jags_model = NULL
)
```

**Arguments**

<code>NAP_prior</code>	An object returned by <code>NAP_prior()</code> that contains the prior specification and (for eNAP) any calibrated tuning parameters $a$ , $b$ .
<code>theta_EC2</code>	Numeric scalar. True log-hazard ratio for E vs C2 used to generate the direct trial data.
<code>n_EC2</code>	Integer. Total sample size for the simulated E vs C2 trial.
<code>lambda</code>	Numeric scalar $> 0$ . Randomization ratio E:C2; e.g., $\lambda = 2$ means 2:1 allocation to E:C2.
<code>sim_model</code>	Character string. Event-time model used to simulate individual times; one of "Exponential" or "Weibull".
<code>model_param</code>	Named numeric vector for the baseline hazard of the control arm. For <code>sim_model = "Exponential"</code> , use <code>c(rate = ...)</code> . For <code>sim_model = "Weibull"</code> , use <code>c(shape = ..., rate = ...)</code> .
<code>iter</code>	Integer. Total MCMC iterations per chain for JAGS (default 2000).
<code>chains</code>	Integer. Number of MCMC chains (default 4).
<code>seed</code>	Integer. Random seed for the simulation replicates.
<code>nsim</code>	Integer. Number of Monte Carlo replicates (default 100).

`jags_model` Either a length-1 character string containing JAGS model code (e.g., a packaged object such as `jags_model_RE`) or a file path to a `.txt` JAGS model. If `NULL`, a default FE/RE model is chosen to match the `NAP_prior` mode.

### Value

A data frame with one row per replicate containing:

- `post_mean`, `post_sd`, `low95`, `hi95` — posterior mean, SD, and 95\
- `prob_E_better` — posterior probability  $\theta_{\{E, C2\}} < 0$ .
- `prior_weight`, `post_weight` — prior and updated weights used in the mixture (for eNAP, `prior_weight` is  $w(Z)$ ).
- `sigma_hat` — posterior mean of between-study SD (RE only; NA for FE).

---

NAP\_posterior

*Conduct posterior inference with NAP-based priors*

---

### Description

Draw posterior via MCMC (JAGS) with derived NAP priors from `NAP_prior` function both setting (one external trial/multiple external trials) and NAP method (NAP/mNAP/eNAP) will be determined by the provided `NAP_prior` object. If using eNAP, make sure the tuning parameter used to derive `NAP_prior` are calibrated by `tune_param_eNAP` function.

### Usage

```
NAP_posterior(
  NAP_prior = NULL,
  y_EC2,
  s_EC2,
  iter = 2000,
  chains = 4,
  model = NULL
)
```

### Arguments

`NAP_prior` An object returned by `NAP_prior()` containing the full the NAP prior (and if eNAP without assumed direct effects, calibrated tuning parameters (a,b))

`y_EC2` Numeric scalar. Direct estimate  $y_{EC2}$  (e.g., log-HR) for  $E$  vs  $C2$ .

`s_EC2` Positive numeric scalar. Sampling variance  $s_{EC2}^2$  for  $y_{EC2}$ .

`iter` Total MCMC iterations per chain (default 2000).

`chains` Number of MCMC chains (default 4).

`model` Either a length-1 character string containing JAGS model code or a path to a JAGS model file. If `NULL`, a package default will be used.

**Value**

A list of class "NAP\_posterior\_result" with elements:

- posterior\_sum: data frame with posterior summaries for  $\theta_{E,C2}$  (mean, sd, 95\ weights (prior\_weight, post\_weight)).
- enap\_prior: For eNAP only: data frame describing the eNAP prior with calculated data-dependent weight: columns for NAP (Informative) and Vague, rows for Mixing Weight, Mean, Variance, and ESS (events) if available.
- jags\_fit: the R2jags fit object.

#'

**Examples**

```
# Create a NAP_prior object
my_naprior <- NAP_prior(
  weight_mtd = "fixed", w = 0.50,      # fixed mixture weight
  y_EC1 = -0.36, s_EC1 = 0.16^2,
  y_C2C1 = -0.30, s_C2C1 = 0.14^2,   # single external trial
  tau0 = 1000
)

# Calculate posterior
out <- NAP_posterior(
  NAP_prior = my_naprior,
  y_EC2 = -0.20, s_EC2 = 0.18^2,
  iter = 1000, chains = 2
)
out$posterior_sum
out$enap_prior
```

---

NAP\_prior

*NAP\_prior: Derive NAP/mNAP/eNAP priors*

---

**Description**

Builds the informative NAP component (mean/variance from the indirect path) and the vague component, and reports the mixing weight depending on the mode:

- weight\_mtd = "fixed": use the supplied fixed weight  $w$  in  $[0, 1]$ .
- weight\_mtd = "adaptive" (eNAP): if  $y_{EC2}$  is provided, compute the data-dependent weight via the elastic link; otherwise, print a formula note.

Derive NAP-based prior (s) based on indirect evidence

Derive the NAP-based posteriors with provided summary statistics on indirect evidence edges By default, the function assumes a vague component is desired, as a result, to obtain NAP/mNAP/eNAP:

- *NAP* Set `weight_mtd="fixed"` and `w=1`, use the NAP (informative component) column results
- *mNAP* Set `weight_mtd="fixed"` and `w` as pre-specified fixed weight. The resulting mNAP is  $w\pi_{NAP} + (1 - w)\pi_0$
- *eNAP* Set `weight_mtd="adaptive"` and provide calibrated `a` and `b` as from `tune_param_eNAP` function, then either: 1). Provide assumed value for `y_EC2` and `s_EC2` (i.e., as for sample size calculation): return a calculated dynamic weight  $w(Z)$ , the resulting eNAP is then  $w(Z)\pi_{NAP} + (1 - w(Z))\pi_0$ ; OR 2). Leave `y_EC2` and `s_EC2` as NULL, return the NAP (informative component) and Vague component, with description for protocol reference

### Usage

```
NAP_prior(
  weight_mtd = c("adaptive", "fixed"),
  w = NULL,
  a = NULL,
  b = NULL,
  y_EC2 = NULL,
  s_EC2 = NULL,
  y_EC1,
  s_EC1,
  y_C2C1,
  s_C2C1,
  mu0 = 0,
  tau0 = 1000,
  lambda = 1,
  sigma2_hat = NULL
)
```

### Arguments

<code>weight_mtd</code>	Either "adaptive" (eNAP) or "fixed" (NP/NAP/mNAP).
<code>w</code>	Fixed prior weight in [0,1]; <b>required only</b> if <code>weight_mtd="fixed"</code> . Ignored otherwise. $0 < w < 1$ infers mixture NAP; $w=0$ infers NP; $w=1$ infers NAP.
<code>a, b</code>	eNAP tuning parameters; <b>required only</b> if <code>weight_mtd="adaptive"</code> ( $a < 0$ and $b > 0$ ). Ignored in fixed mode.
<code>y_EC2, s_EC2</code>	Log-HR and SE for $E : C2$ (Current trial post-SoC change).
<code>y_EC1, s_EC1</code>	Log-HR and SE for $E : C1$ (Current trial pre-SoC change).
<code>y_C2C1, s_C2C1</code>	Historical C2 vs. C1 trial Log-HRs and SEs.
<code>mu0, tau0</code>	mean and variance of the vague component (default $\sqrt{1000}$ ).
<code>lambda</code>	Randomization ratio (default 1).
<code>sigma2_hat</code>	Positive scalar, required only for multiple external trials setting, leave blank if use default REML estimate, otherwise provide user-specified value

### Details

This function automatically selects one external trial vs multiple external trials setting:

- One external trial if  $\text{length}(y\_C2C1) == 1$  &  $\text{length}(s\_C2C1) == 1$  (one external trial).
- Multiple external trials if  $\text{length}(y\_C2C1) > 1$  &  $\text{length}(s\_C2C1) == \text{length}(y\_C2C1)$ . By default uses `metafor::rma.uni(..., method="REML")` to obtain REML estimate; Otherwise please provide `sigma2_hat`

### Value

Displays the NAP prior as a mixture of an informative prior (constructed based on the indirect evidence path) and a vague prior.

An object of class "NAPrior" (data.frame + attributes).

### Examples

```
## -----
## Example 1: One external trial setting with fixed mixing weight of 0.5 (mNAP)
## -----
mNAP_test1 <- NAP_prior(
  weight_mtd = "fixed", w = 0.50,           # fixed mixture weight
  y_EC1 = -0.36, s_EC1 = 0.16^2,
  y_C2C1 = -0.30, s_C2C1 = 0.14^2,       # single external trial
  tau0 = 1000
)
print(mNAP_test1)
plot(mNAP_test1)

## -----
## Example 2: RE case (multiple historical), ADAPTIVE weight
## -----
eNAP_test1 <- NAP_prior(
  weight_mtd = "adaptive",
  a = -2, b = 10,                         # from calibration
  y_EC1 = -0.36, s_EC1 = 0.16^2,         # E:C1 (current, pre-change)
  y_C2C1 = c(-0.28, -0.35, -0.31),     # C2:C1 (external trials)
  s_C2C1 = c(0.12^2, 0.11^2, 0.15^2),
  tau0 = 1000                             # vague variance
)
print(eNAP_test1)
```

---

post\_w

*Posterior mixture weight calculation*

---

### Description

Computes posterior updated mixture weights for a two-component normal-normal model using the standard logit-additive update. The *prior* mixing weight is either a fixed weight  $w \in (0, 1)$  or a dynamic mixing weight as for eNAP prior:  $Z = n_{eff}^{-1/4} |y_{dir} - y_{ind}| / s_{link}$ : where,  $n_{eff} = s_{link}^{-1}$   
 $w(Z) = 1 / \exp(a + b \log(Z + 1))$ ,  $a < 0$ ,  $b > 0$ .

**Usage**

```

post_w(
  w,
  a,
  b,
  s_EC2,
  s_EC1,
  s_C2C1,
  y_EC2,
  y_EC1 = -0.5,
  y_C2C1 = -0.5,
  tau0 = 1000,
  mu0 = 0,
  eps = 1e-12
)

```

**Arguments**

w	Scalar. If $w > 1$ , use the ADAPTIVE branch (logistic prior on log-Z). If $0 < w < 1$ , use a fixed prior weight equal to w.
a, b	Parameters used in the elastic function for dynamic mixing weight. Must satisfy $a < 0$ and $b > 0$ .
s_EC2, s_EC1, s_C2C1	Sampling variances for direct evidence (E vs. C2 trial), and edges of indirect evidence (E vs. C1 trial and C2 vs. C1 trial).
y_EC2, y_EC1, y_C2C1	Estimated log-HR for E vs. C2 trial, E vs. C1 trial, C2 vs. C1 trial, respectively
mu0, tau0	Mean and variance for the vague component.
eps	Numeric scalar used for small-value clipping (default 1e-12).

**Details**

- **Fixed prior mixing weight (Robust NMAP Prior):** requires  $0 < w < 1$ .
- **Adaptive branch (Adaptive NMAP Prior):** triggered by  $w > 1$ , requires  $a < 0$  and  $b > 0$ . This corresponds to a decreasing prior weight as the inconsistency grows.
- All variance/SD arguments may be given as scalars or vectors; scalars are recycled.

**Value**

A numeric vector of posterior weights in  $(0, 1)$  reflecting realized borrowing fraction of the informative component.

**Examples**

```

y_EC2 <- -0.5; y_EC1 <- -0.8; y_C2C1 <- -0.3
s_EC2 <- 0.2; s_EC1 <- 0.18; s_C2C1 <- 0.18

```

```
# Fixed mixing weight 0.5
post_w(w = 0.5, a = NA, b = NA, s_EC2, s_EC1, s_C2C1,
       y_EC2, y_EC1, y_C2C1)

# Dynamic weight with elastic function of (a=-4.6, b=3):
post_w(w = 2, a = -2.5, b = 10, s_EC2, s_EC1, s_C2C1,
       y_EC2, y_EC1, y_C2C1)
```

---

tune_param_eNAP	<i>Calibrate (a, b) for eNAP prior</i>
-----------------	--

---

## Description

Calibrates the tuning parameters ( $a$ ,  $b$ ) of the elastic NAP prior. This function supports both the one external trial setting and multiple external trials setting:

- *Single external trial* provide  $y_{C2C1}$  and  $s_{C2C1}$  as scalars.
- *Multiple external trials* provide  $y_{C2C1}$  and  $s_{C2C1}$  as vectors of same lengths. by default the cross-trial variance will be automatically calculated by REML, otherwise please provide the cross-trial variance as input parameter:  $\sigma^2_{\hat{}}$

## Usage

```
tune_param_eNAP(
  s_EC2,
  s_EC1,
  s_C2C1,
  tau0 = 1000,
  delta = 0.5,
  t1 = 0.999,
  t0 = 0.05,
  clip_a = c(-5, -0.5),
  clip_b = c(1e-05, 50),
  exact = FALSE,
  y_EC1 = -0.5,
  y_C2C1 = -0.5,
  mu0 = 0,
  sigma2_hat = NULL,
  verbose = FALSE
)
```

## Arguments

$s_{EC2}$ ,  $s_{EC1}$ ,  $s_{C2C1}$

Sampling variances for post-SoC change period (E vs. C2), pre-SoC change period of current trial (E vs. C1 trial) and external trial (C2 vs. C1 trial)

delta	Positive scalar; Clinically significant difference on the log-HR scale such that direct and indirect evidence should be considered as strongly inconsistent.
t1, t0	Positive scalar; Calibration targets at consistency and strongly inconsistency: $w'(0) = t1$ (near 1; default 0.99), $w'(\delta) = t0$ (near 0; default 0.05).
clip_a, clip_b	Numeric Vector of Length 2: Minimum and maximum caps for tuning parameters (a,b), by default clip_a=(-5,0.5) and clip_b=(0,50)
exact	Logical (TRUE/FALSE); If TRUE, require the exact solution for parameter (a,b), which further requires more parameters input
y_EC1, y_C2C1	Log-HR for pre-SoC change period and external trial, required only if exact=TRUE
mu0, tau0	Mean and variance for the vague component, by default mu0=0 and tau0=1000.
sigma2_hat	Positive scalar, required only for multiple external trials setting, leave blank if use default REML estimate, otherwise provide user-specified value
verbose	Logical; print diagnostics.

## Details

### Calibration procedure:

- *Consistency case* ( $Z = 0$ ). Enforce near-full borrowing at exact consistency by solving  $w'(Z = 0) = t_1$  for  $a$ .
- *Strong inconsistency case* ( $Z(\delta) = \frac{|\delta|}{\sqrt{s_{E,C_2} + s_{E,C_1} + s_{C_2,C_1}}}$ ). Enforce minimal borrowing at a clinically significant difference by targeting the *updated* weight  $w'(Z(\delta)) = t_0$ , with calibrated  $a$  from step 1, solve for  $b$ .

For further details, see the original NAP paper by Zhang and et al. (manuscript).

## Value

list with a, b, mode ("FE" or "RE"), and simple check summary.

## Examples

```
s_EC2 <- 0.2^2; s_EC1 <- 0.18^2; s_C2C1 <- 0.18^2
tau0 <- 1000

# One external trial setting
tune_param_eNAP(
  s_EC2,s_EC1,s_C2C1, tau0=1000,
  delta=0.5, t1 = 0.999, t0 = 0.05)

# Multiple external trials setting
s_C2C1=c(0.19^2,0.18^2,0.20^2)
y_C2C1=c(-0.5,-0.45,-0.6)
tune_param_eNAP(
  s_EC2,s_EC1,s_C2C1, tau0=10,
  delta=0.5, t1 = 0.999, t0 = 0.05,
  exact=TRUE,y_EC1=-0.8,y_C2C1=y_C2C1)
```

# Index

NAP\_oc, [2](#)

NAP\_posterior, [3](#)

NAP\_prior, [2](#), [3](#), [4](#)

post\_w, [6](#)

tune\_param\_eNAP, [8](#)