

Package: rbpengine (via r-universe)

October 7, 2026

Title Relevance-Based Prediction for the 'RBP Math Engine'

Version 2.4.0

Description Provides predict_psr(), predict_maxfit(), and predict_grid() for relevance-based prediction ('RBP') via the 'RBP Math Engine' C application binary interface ('ABI'). Returns a 'PredictionResults' nested list that mirrors the 'Python' 'rbp_engine' package. The package builds and installs without the private engine; prediction and insight entry points load the engine at runtime ('RBP_ENGINE_LIB' or a staged install under 'RBP_ENGINE_HOME' / /Library/rbp-engine). Licensing uses 'rbp-license-info' and standard license environment variables.

License MIT + file LICENSE

Encoding UTF-8

Depends R (>= 4.1.0)

Imports utils, jsonlite

Suggests testthat (>= 3.0.0), digest

Config/testthat/edition 3

SystemRequirements Optional RBP Math Engine runtime (shared library librbp_math_lib plus platform BLAS/LAPACK companions and license tools). Not required to install or load this package; required for prediction.

URL <https://cambridgesportsanalytics.r-universe.dev>,
<https://docs.csanalytics.io/Install/R>,
<https://github.com/CambridgeSportsAnalytics/rbp-engine-r>

BugReports <https://github.com/CambridgeSportsAnalytics/rbp-engine-r/issues>

Roxygen list(markdown = TRUE)

Config/roxygen2/version 8.1.0

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

Date/Publication 2026-10-05 21:47:49 UTC

```
RemoteUrl https://github.com/CambridgeSportsAnalytics/rbp-engine-r
RemoteRef HEAD
RemoteSha 01dff44862c005dbedcdad1afabdc0d486e0a35b
```

Contents

default_engine_home	2
engine	3
engine_load	4
engine_releases_url	4
engine_runtime_id	5
engine_unload	5
ensure_engine	6
ensure_license_env	6
GridOptions	7
install_engine	9
license_info	10
license_source	10
math_lib_version	11
predict_grid	11
predict_maxfit	12
predict_psr	13
PredictOptions	14
print.PredictionResults	16
relevance	16
set_percentile_value_algorithm	17
uninstall_engine	18
version-helpers	18
Index	20

default_engine_home	Default prefix used by install_engine() .
---------------------	---

Description

Default prefix used by [install_engine\(\)](#).

Usage

```
default_engine_home()
```

Value

A directory path.

Examples

```
default_engine_home()
```

engine	<i>Locate and load the private RBP Math Engine runtime.</i>
--------	---

Description

The open **rbpengine** package is a thin client. Predictions require the private **RBP Math Engine** shared library (`librbp_math_lib.dylib` / `.so` / `rbp_math_lib.dll`) plus platform companions (OpenBLAS/LAPACK where needed) and a valid license.

Usage

```
engine_candidate_paths()
```

```
engine_available()
```

```
find_engine_path()
```

Details

Discovery order (first path that loads and exposes the expected ABI symbols):

1. `RBP_ENGINE_LIB` (full path to the shared library)
2. `RBP_ENGINE_HOME` / `lib/` (+ OS filename)
3. `install_engine()` prefix (`default_engine_home()`)
4. Well-known install prefixes (`/Library/rbp-engine/lib`, `$HOME/.local/share/rbp-engine/lib`, ...)
5. Developer cargo tree (`RBP_MATH_LIB_ROOT`, sibling `rbp-math-lib/target`, or `../target` when this package still sits next to a cargo workspace)

Value

`engine_available()` returns a logical. `engine_candidate_paths()` returns character paths that may be probed. `find_engine_path()` and `engine_load()` return the loaded path (invisibly). `engine_unload()` returns `TRUE` invisibly after releasing the loaded library. `ensure_engine()` returns `TRUE` invisibly when the engine is available.

See Also

`library_path()`, `abi_version()`, `install_engine()`

Examples

```
engine_available()
length(engine_candidate_paths()) >= 1
```

<code>engine_load</code>	<i>Load the engine shared library at an explicit path.</i>
--------------------------	--

Description

Load the engine shared library at an explicit path.

Usage

```
engine_load(path = NULL)
```

Arguments

<code>path</code>	Full path to <code>librbp_math_lib.*</code> (or set <code>RPB_ENGINE_LIB</code> and call <code>engine_load()</code> with no args after relying on <code>find_engine_path()</code>).
-------------------	--

Value

Invisibly, the path loaded.

<code>engine_releases_url</code>	<i>Public HTTPS prefix for runtime tarballs.</i>
----------------------------------	--

Description

Override with env `RPB_ENGINE_RELEASES_URL` (no trailing slash).

Usage

```
engine_releases_url()
```

Value

Character scalar.

Examples

```
engine_releases_url()
```

engine_runtime_id	<i>Platform id matching manifest.json artifacts (macos-arm64, ...).</i>
-------------------	---

Description

Platform id matching manifest.json artifacts (macos-arm64, ...).

Usage

```
engine_runtime_id()
```

Value

Character scalar.

Examples

```
engine_runtime_id()
```

engine_unload	<i>Release a loaded engine shared library.</i>
---------------	--

Description

[install_engine\(\)](#) calls this before replacing a runtime that this session already has mapped, so the file is not locked for the copy.

Usage

```
engine_unload()
```

Value

Invisibly TRUE.

<code>ensure_engine</code>	<i>Ensure the engine is loaded; stop with install instructions if not.</i>
----------------------------	--

Description

Called by prediction helpers. Safe to call repeatedly.

Usage

```
ensure_engine()
```

Value

Invisibly TRUE when available.

<code>ensure_license_env</code>	<i>Adopt a default on-disk license when the process environment has none.</i>
---------------------------------	---

Description

Matches the Python `rbp_engine._license` resolution order:

1. RBP_LICENSE
2. RBP_LICENSE_FILE
3. `~/.rbp-engine/license.txt`
4. `/Library/rbp-engine/license.txt` (macOS system install)

Usage

```
ensure_license_env()
```

Details

Never errors. Called automatically from `.onLoad`.

Value

The path adopted (invisibly), or NULL when the environment already carries a license or no default file exists.

GridOptions

Grid search options for [predict_grid](#).

Description

Same censor / scale / inversion knobs as [PredictOptions](#), plus combination search settings. Does **not** take `verify_missing_data` / `missing_moments`: Grid chooses complete vs pairwise moments per combination.

Usage

```
GridOptions(
  threshold = c(0, 0.2, 0.5, 0.8),
  censor_type = "both",
  censor_unit = "percent",
  censor_operator = "greater_than_or_equal_to",
  prediction_scale = "response",
  adj_fit_multiplier = "k",
  inv_method = "gaussian",
  include_linear_regression = FALSE,
  verbose = FALSE,
  max_iter = 1000L,
  min_k = NULL,
  max_k = NULL,
  allowed_k = NULL,
  k = 1L,
  seed = 42L,
  retain_all = FALSE,
  retain_grid_objects = NULL,
  attribute_combi = NULL,
  required_attributes = NULL,
  attribute_groups = NULL,
  adjust_impact_for_missing = TRUE,
  inner_parallel = "auto"
)
```

Arguments

<code>threshold</code>	Numeric vector of thresholds. Default <code>c(0, 0.2, 0.5, 0.8)</code> .
<code>censor_type</code>	"relevance", "similarity", or "both" (default).
<code>censor_unit</code>	"percent" or "score".
<code>censor_operator</code>	Comparison vs threshold. Default "greater_than_or_equal_to".
<code>prediction_scale</code>	"response" or "logistic".

<code>adj_fit_multiplier</code>	"k", "identity", or "log".
<code>inv_method</code>	"gaussian", "cholesky", or "pseudoinverse".
<code>include_linear_regression</code>	Logical; fills <code>yhat_linear</code> when available.
<code>verbose</code>	Logical; engine diagnostics (default <code>FALSE</code>).
<code>max_iter</code>	Combination search budget.
<code>min_k</code>	Minimum number of attributes turned on in a combination. Combinations may be larger, up to <code>max_k</code> or all attributes, until <code>max_iter</code> is filled. Default 1.
<code>max_k</code>	Maximum number of attributes turned on. Omit it to allow every attribute. Set it equal to <code>min_k</code> for that exact size.
<code>allowed_k</code>	Integer vector of exact combination sizes, such as <code>c(1L, 3L, 7L, 10L)</code> . Replaces <code>min_k</code> and <code>max_k</code> .
<code>k</code>	Alias of <code>min_k</code> . Still accepted.
<code>seed</code>	RNG seed.
<code>retain_all</code>	Retain all grid cell objects.
<code>retain_grid_objects</code>	Character policy string (overrides <code>retain_all</code>).
<code>attribute_combi</code>	Optional matrix of attribute combinations. Cannot be combined with <code>required_attributes</code> or <code>attribute_groups</code> .
<code>required_attributes</code>	Length-K 0/1 mask of attributes that are on in every generated combination.
<code>attribute_groups</code>	G by K 0/1 mask. Each row is a group of attributes selected together.
<code>adjust_impact_for_missing</code>	Logical; incomplete-column IOF/IOP vs a + Z include-k null (default <code>TRUE</code>). <code>FALSE</code> is the pre-adjustment / PSR-parity baseline.
<code>inner_parallel</code>	"auto" or "off" (within-call parallelism).

Value

A named list with class `GridOptions`.

See Also

[PredictOptions](#), [predict_grid](#)

Examples

```
GridOptions(min_k = 2L, max_iter = 100L)
```

install_engine	<i>Install the private RBP Math Engine runtime.</i>
----------------	---

Description

The open **rbpengine** package is a thin client. Predictions need the private engine (shared library + companions + **rbp-license-info**). This helper fetches the platform tarball from <https://docs.csanalytics.io/releases/> (or unpacks a local archive) into a well-known prefix that [engine_candidate_paths\(\)](#) already searches.

Usage

```
install_engine(
  version = "latest",
  dest = NULL,
  url = NULL,
  path = NULL,
  load = TRUE,
  quiet = FALSE
)
```

Arguments

version	Math runtime to fetch (docs.csanalytics.io/releases). "latest" (default) or a math-lib semver such as "1.3.7". Ignored when path or url is set.
dest	Install prefix. Default is default_engine_home() (<code>~/.local/share/rbp-engine</code> on Unix, <code>%LOCALAPPDATA%/rbp-engine</code> on Windows).
url	Full URL of a <code>.tar.gz</code> (skips the manifest). <code>sha256</code> is not verified unless manifest_url is also reachable via version .
path	Local <code>.tar.gz</code> or an already-staged runtime directory (<code>lib/</code> , <code>bin/</code>).
load	If TRUE (default), engine_load() the newly installed library.
quiet	Suppress download progress.

Details

This is a **user-called** step, never run at package install time and never invoked from examples, tests, or vignettes (CRAN policy). Analogous to `pip install rbp-engine` for the engine files only. Typical calls are `install_engine()` for the latest math runtime for this machine, `install_engine(version = "1.3.7")` to pin an rbp-math-lib tarball (not the PyPI **rbp-engine** 2.x version), or `install_engine(path = "macos-arm64.tar.gz")` for a local archive. Calling it again upgrades in place. A library this session already loaded from **dest** is unloaded first. A file another process still has mapped is renamed aside and the new file is written under the original name, because a loaded shared library cannot be overwritten in place.

Value

The install prefix, invisibly.

See Also

`ensure_engine()`, `engine_available()`, `uninstall_engine()`

<code>license_info</code>	<i>Run the <code>rbp-license-info</code> CLI shipped with the engine runtime.</i>
---------------------------	---

Description

Looks in the installed runtime `bin/` (see `install_engine()`), well-known prefixes, cargo `target/`, then `PATH`.

Usage

`license_info(...)`

Arguments

... Arguments passed to the binary (character; e.g. `--viewlicense`).

Value

Exit status (invisibly).

<code>license_source</code>	<i>Human-readable description of where the license is coming from.</i>
-----------------------------	--

Description

Human-readable description of where the license is coming from.

Usage

`license_source()`

Value

A short character string such as `"RBP_LICENSE (environment)"`, `"RBP_LICENSE_FILE=..."`, or a message indicating none is configured.

Examples

`license_source()`

<code>math_lib_version</code>	<i>rbp-math-lib semver embedded in the shared library on this machine.</i>
-------------------------------	--

Description

Reads the `RBP-ENGINE-VERSION=` record compiled into the engine dll, dylib, or so. This does not read package, config, or version files, and it does not load the library, so a missing BLAS companion or a wrong-architecture binary can still report the version baked into that file.

Usage

```
math_lib_version(path = NULL)
```

Arguments

<code>path</code>	Optional full path to a shared library. When <code>NULL</code> , the library on this system is located as described above.
-------------------	--

Details

When `path` is omitted, `RBP_ENGINE_LIB` is the only file inspected if it is set. Otherwise each existing file from `engine_candidate_paths()` is read until one contains the record. A binary without that record is skipped, so an older install does not hide a newer build. The filename depends on the OS: `librbp_math_lib.dylib` (macOS), `librbp_math_lib.so` (Linux), or `rbp_math_lib.dll` (Windows).

Prints one line to the terminal, `rbp-math-lib <version> (<path>)`.

Value

The semver string (for example "1.3.7"), invisibly.

See Also

`engine_candidate_paths()`, `abi_version()`, `package_version_rbpengine()`

<code>predict_grid</code>	<i>Fit-weighted composite across attribute combinations.</i>
---------------------------	--

Description

Evaluates combinations of predictive attributes across relevance and similarity thresholds, then forms a fit-weighted composite prediction for circumstances `theta`. Combinations that fit this task more reliably pull harder on the forecast. The result also includes variable-level Impact on Fit and Impact on Prediction.

Usage

```
predict_grid(y, X, theta, options = NULL)
```

Arguments

y Numeric outcomes, length N.
X Numeric matrix, N rows by K columns (R column-major is native).
theta Circumstances vector, length K.
options Optional [PredictOptions](#), [MaxFitOptions](#), or [GridOptions](#).

Details

Recommended default path for analysis.

Value

A nested list of class `PredictionResults` (same field layout as Python `rbp_engine.PredictionResults`). Access fields with `$`, e.g. `result$yhat`, `result$insights$relevance`.

Examples

```
PredictOptions(threshold = 0.5)
if (engine_available()) {
  set.seed(1)
  N <- 40; K <- 3
  X <- matrix(rnorm(N * K), N, K)
  y <- rnorm(N)
  theta <- colMeans(X)
  out <- predict_grid(y, X, theta, GridOptions(k = 1L, max_iter = 50L))
  as.numeric(out$yhat[1])
  best <- predict_maxfit(y, X, theta)
  res <- predict_psr(y, X, theta, PredictOptions(threshold = 0.5))
  res$yhat
}
```

`predict_maxfit`

Strongest prediction on a fixed set of attributes.

Description

Searches candidate thresholds and returns the single winner on your objective: fit, adjusted fit, or k-fit (the default).

Usage

```
predict_maxfit(y, X, theta, options = NULL)
```

Arguments

<code>y</code>	Numeric outcomes, length N.
<code>X</code>	Numeric matrix, N rows by K columns (R column-major is native).
<code>theta</code>	Circumstances vector, length K.
<code>options</code>	Optional PredictOptions , MaxFitOptions , or GridOptions .

Details

Useful when you want that maximum on one attribute set, without a full combination grid. For primary analysis, prefer [predict_grid\(\)](#), which diversifies across attribute combinations and thresholds, then blends by fit.

Value

A nested list of class `PredictionResults`. `yhat`, `fit`, and `thresholds` have length 1 (the winner).

Examples

```
if (engine_available()) {
  set.seed(1)
  N <- 40; K <- 3
  X <- matrix(rnorm(N * K), N, K)
  y <- rnorm(N)
  theta <- colMeans(X)
  best <- predict_maxfit(y, X, theta)
  best$yhat
  best <- predict_maxfit(y, X, theta, MaxFitOptions(objective = "fit"))
}
```

`predict_psr`

Partial-sample regression at thresholds you choose.

Description

Forms a relevance-weighted forecast at the censoring thresholds you choose. The prediction uses the informative, relevant subset of the sample, rather than treating every past observation equally.

Usage

```
predict_psr(y, X, theta, options = NULL)
```

Arguments

<code>y</code>	Numeric outcomes, length N.
<code>X</code>	Numeric matrix, N rows by K columns (R column-major is native).
<code>theta</code>	Circumstances vector, length K.
<code>options</code>	Optional PredictOptions , MaxFitOptions , or GridOptions .

Details

This is the evaluation inside each Grid cell and inside MaxFit. Use `predict_psr()` when the attribute set and thresholds are already fixed. For primary analysis, prefer [predict_grid\(\)](#).

Named `predict_psr` (not `predict`) so base R's `stats::predict` is not masked.

Value

A nested list of class `PredictionResults`. `yhat` has one prediction per threshold.

Examples

```
if (engine_available()) {
  set.seed(1)
  N <- 40; K <- 3
  X <- matrix(rnorm(N * K), N, K)
  y <- rnorm(N)
  theta <- colMeans(X)
  res <- predict_psr(y, X, theta, PredictOptions(threshold = 0.5))
  res$yhat
}
```

PredictOptions	<i>Configuration objects (named lists) for predict / maxfit / grid.</i>
----------------	---

Description

Mirrors Python `PredictOptions`, `MaxFitOptions`, and `GridOptions`. Pass the list to [predict_psr](#), [predict_maxfit](#), or [predict_grid](#). Fields set to NULL are omitted so the engine applies its defaults.

Usage

```
PredictOptions(
  threshold = NULL,
  censor_type = "relevance",
  censor_unit = "percent",
  censor_operator = "greater_than_or_equal_to",
  prediction_scale = "response",
  adj_fit_multiplier = "k",
```

```

    inv_method = "gaussian",
    verify_missing_data = FALSE,
    include_linear_regression = FALSE,
    verbose = FALSE
  )

  MaxFitOptions(
    threshold = c(0, 0.2, 0.5, 0.8),
    censor_type = "both",
    censor_unit = "percent",
    censor_operator = "greater_than_or_equal_to",
    prediction_scale = "response",
    adj_fit_multiplier = "k",
    inv_method = "gaussian",
    verify_missing_data = FALSE,
    include_linear_regression = FALSE,
    verbose = FALSE,
    objective = "kfit",
    inner_parallel = "auto"
  )

```

Arguments

threshold	Numeric vector of thresholds, or NULL for engine default.
censor_type	Character: "relevance", "similarity", or "both" ("both" only for maxfit/grid).
censor_unit	"percent" or "score".
censor_operator	Comparison vs threshold. Default "greater_than_or_equal_to" (">=").
prediction_scale	"response" or "logistic".
adj_fit_multiplier	"k", "identity", or "log".
inv_method	"gaussian", "cholesky", or "pseudoinverse".
verify_missing_data	Logical. Predict / MaxFit only (not GridOptions).
include_linear_regression	Logical; fills <code>yhat_linear</code> when available.
verbose	Logical; when TRUE, the engine may print library status diagnostics (default FALSE).
objective	MaxFit objective: "kfit", "fit", or "adjusted_fit".
inner_parallel	"auto" or "off" (within-call parallelism).

Value

A named list with class `PredictOptions` or `MaxFitOptions`.

See Also

[GridOptions](#), [predict_psr](#), [predict_maxfit](#)

Examples

```
PredictOptions(threshold = 0.5)
MaxFitOptions(objective = "kfit")
```

```
print.PredictionResults
```

Print a compact summary of prediction results.

Description

Print a compact summary of prediction results.

Usage

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

Arguments

<code>x</code>	A <code>PredictionResults</code> object from predict_grid , predict_maxfit , or predict_psr .
<code>...</code>	Unused.

Value

`x`, invisibly.

`relevance`

Standalone insight scores (no prediction call required).

Description

These call the loaded RBP Math Engine directly. Use [ensure_engine\(\)](#) / [engine_available\(\)](#) if you need to check the runtime first.

Usage

```

relevance(X, theta)

similarity(X, theta)

info_x(X)

info_theta(X, theta)

relevance_metrics(X, theta)

```

Arguments

X Numeric matrix, N rows by K columns.

theta Circumstances vector, length K (not used by [info_x\(\)](#)).

Value

For [relevance\(\)](#) and [similarity\(\)](#), a numeric vector of length N. [info_x\(\)](#) returns a numeric vector of length K. [info_theta\(\)](#) returns a single numeric. [relevance_metrics\(\)](#) returns a named list of metric vectors/scalars from the engine.

Examples

```

X <- matrix(rnorm(20 * 3), 20, 3)
theta <- colMeans(X)
if (engine_available()) {
  relevance(X, theta)
  similarity(X, theta)
  info_x(X)
  info_theta(X, theta)
  relevance_metrics(X, theta)
}

```

```
set_percentile_value_algorithm
```

*Process-wide percentile cut algorithm used when **censor_unit** is percent.*

Description

Process-wide percentile cut algorithm used when **censor_unit** is percent.

Usage

```

set_percentile_value_algorithm(algorithm)

get_percentile_value_algorithm()

```

Arguments

`algorithm` Integer code or string naming the algorithm (for example "full_sort", "order_statistics"). Exact accepted values are defined by the loaded engine ABI.

Value

`set_percentile_value_algorithm()` returns NULL invisibly. `get_percentile_value_algorithm()` returns the current algorithm code as an integer.

<code>uninstall_engine</code>	<i>Remove a runtime tree installed by <code>install_engine()</code>.</i>
-------------------------------	--

Description

Refuses to delete a directory that does not look like an RBP runtime (`lib/librbp_math_lib.*` or `lib/rbp_math_lib.dll` present).

Usage

```
uninstall_engine(dest = NULL)
```

Arguments

`dest` Prefix to remove. Default `default_engine_home()`.

Value

TRUE invisibly if removed.

<code>version-helpers</code>	<i>Package / ABI version and path diagnostics.</i>
------------------------------	--

Description

Package / ABI version and path diagnostics.

Usage

```
package_version_rbpengine()
```

```
abi_version()
```

```
engine_path_loaded()
```

```
library_path()
```

Value

`package_version_rbpengine()` returns the installed package version as a string. `abi_version()` returns the loaded engine C ABI version as an integer (errors if the engine is not loaded). `engine_path_loaded()` returns the resolved engine library path, or `NA_character_` when none is loaded. `library_path()` returns a named list of engine/license diagnostics including candidate search paths.

See Also

`engine_available()`, `ensure_engine()`, `license_source()`, `math_lib_version()`

Examples

```
package_version_rbpengine()
engine_path_loaded()
```

Index

`abi_version` (*version-helpers*), 18
`abi_version()`, 3, 11, 19

`default_engine_home`, 2
`default_engine_home()`, 3, 9, 18

`engine`, 3
`engine_available` (*engine*), 3
`engine_available()`, 3, 10, 16, 19
`engine_candidate_paths` (*engine*), 3
`engine_candidate_paths()`, 3, 9, 11
`engine_load`, 4
`engine_load()`, 3, 4, 9
`engine_path_loaded` (*version-helpers*), 18
`engine_path_loaded()`, 19
`engine_releases_url`, 4
`engine_runtime_id`, 5
`engine_unload`, 5
`engine_unload()`, 3
`ensure_engine`, 6
`ensure_engine()`, 3, 10, 16, 19
`ensure_license_env`, 6

`find_engine_path` (*engine*), 3
`find_engine_path()`, 3, 4

`get_percentile_value_algorithm` (*set_percentile_value_algorithm*), 17
`get_percentile_value_algorithm()`, 18
`GridOptions`, 7, 12–16

`info_theta` (*relevance*), 16
`info_theta()`, 17
`info_x` (*relevance*), 16
`info_x()`, 17
`install_engine`, 9
`install_engine()`, 2, 3, 5, 10, 18

`library_path` (*version-helpers*), 18
`library_path()`, 3, 19
`license_info`, 10
`license_source`, 10
`license_source()`, 19

`math_lib_version`, 11
`math_lib_version()`, 19
`MaxFitOptions`, 12–14
`MaxFitOptions` (*PredictOptions*), 14

`package_version_rbpengine` (*version-helpers*), 18
`package_version_rbpengine()`, 11, 19
`predict_grid`, 7, 8, 11, 14, 16
`predict_grid()`, 13, 14
`predict_maxfit`, 12, 14, 16
`predict_psr`, 13, 14, 16
`PredictOptions`, 7, 8, 12, 13, 14, 14
`print.PredictionResults`, 16

`relevance`, 16
`relevance()`, 17
`relevance_metrics` (*relevance*), 16
`relevance_metrics()`, 17

`set_percentile_value_algorithm`, 17
`set_percentile_value_algorithm()`, 18
`similarity` (*relevance*), 16
`similarity()`, 17
`stats::predict`, 14

`uninstall_engine`, 18
`uninstall_engine()`, 10

`version-helpers`, 18