

Package: writeAlizer (via r-universe)

October 19, 2026

Type Package

Title Generate Predicted Writing Quality Scores

Version 1.7.4

Date 2026-09-16

Description Imports variables from 'ReaderBench' (Dascalu et al., 2018)<[doi:10.1007/978-3-319-66610-5_48](https://doi.org/10.1007/978-3-319-66610-5_48)>, 'Coh-Metrix' (McNamara et al., 2014)<[doi:10.1017/CBO9780511894664](https://doi.org/10.1017/CBO9780511894664)>, and/or 'GAMET' (Crossley et al., 2019) <[doi:10.17239/jowr-2019.11.02.01](https://doi.org/10.17239/jowr-2019.11.02.01)> output files; downloads predictive scoring models documented in Keller-Margulis et al. (2026)<[doi:10.35542/osf.io/36dn5_v1](https://doi.org/10.35542/osf.io/36dn5_v1)>, Mercer & Cannon (2022)<[doi:10.31244/jero.2022.01.03](https://doi.org/10.31244/jero.2022.01.03)>, and Mercer et al. (2021)<[doi:10.1177/0829573520987753](https://doi.org/10.1177/0829573520987753)>; and generates predicted writing quality and curriculum-based measurement (McMaster & Espin, 2007)<[doi:10.1177/00224669070410020301](https://doi.org/10.1177/00224669070410020301)> scores.

License MIT + file LICENSE

URL <https://github.com/shmercercer/writeAlizer/>,
<https://shmercercer.github.io/writeAlizer/>

BugReports <https://github.com/shmercercer/writeAlizer/issues>

Depends R (>= 4.0.0)

Imports caret, cli, digest, dplyr, glue, magrittr, rlang (>= 1.1.0), stats, tidyselect

Suggests caretEnsemble, Cubist, curl, earth, gbm, glmnet, kernlab, knitr, pls, randomForest, rmarkdown, testthat (>= 3.2.0), withr

Config/testthat/edition 3

Encoding UTF-8

Language en-US

VignetteBuilder knitr

Config/roxygen2/version 8.1.0

Config/pak/sysreqs libicu-dev

Repository <https://shmercer.r-universe.dev>
Date/Publication 2026-10-19 19:06:23 UTC
RemoteUrl <https://github.com/shmercer/writealizer>
RemoteRef HEAD
RemoteSha cc8a9d404cae1ff5d8ab1a7965a16dfb2b0f9cc7

Contents

writeAlizer-package	2
import_coh	3
import_gamet	4
import_merge_gamet_rb	5
import_rb	6
keep_stem_before_txt	7
model_deps	8
predict_quality	9
preprocess	10
wa_cache_clear	11
wa_cache_dir	12
wa_download	13
wa_seed_example_models	14
Index	16

writeAlizer-package	<i>writeAlizer: An R Package to Generate Automated Writing Quality and Curriculum-Based Measurement (CBM) Scores.</i>
---------------------	---

Description

Package-level documentation for writeAlizer.

Details

Detailed documentation on writeAlizer is available in the [GitHub README file](#) and [getting-started guide](#).

The writeAlizer R package (a) imports [ReaderBench](#), [Coh-Metrix](#), and [GAMET](#) output files into R, and (b) uses research-developed scoring models to generate predicted writing quality scores or Total Words Written, Words Spelled Correctly, Correct Word Sequences, and Correct Minus Incorrect Word Sequences scores from GAMET files.

The writeAlizer package includes functions to do two types of tasks: (1) importing ReaderBench, Coh-Metrix, and/or GAMET output files into R; and (2) generating predicted quality scores using the imported output files. There are also additional functions to help with (3) identifying optional package dependencies and (4) cache management.

1. Import output files

- [import_rb](#)
- [import_coh](#)
- [import_gamet](#)
- [import_merge_gamet_rb](#)

2. Generate predicted quality scores

- [predict_quality](#)

3. Identify necessary packages

- [model_deps](#)

4. Cache management

- [wa_cache_dir](#)
- [wa_cache_clear](#)

Author(s)

Maintainer: Sterett H. Mercer <sterett.mercer@ubc.ca> ([ORCID](#))

Authors:

- Sterett H. Mercer <sterett.mercer@ubc.ca> ([ORCID](#))

See Also

Useful links:

- <https://github.com/shmercer/writeAlizer/>
- <https://shmercer.github.io/writeAlizer/>
- Report bugs at <https://github.com/shmercer/writeAlizer/issues>

import_coh

Import a Coh-Metrix output file (.csv) into R.

Description

Import a Coh-Metrix output file (.csv) into R.

Usage

import_coh(path)

Arguments

path A string giving the path and filename to import. Keep the original CSV column names. IDs must be nonblank and unique; leading zeros are preserved. Rows are sorted by ID.

Value

A base `data.frame` with one row per record and the following columns:

- ID (character): unique identifier of the text/essay.
- One column per retained Coh-Metrix feature, kept by original feature name (numeric). Feature names mirror the Coh-Metrix output variables.

The object has class `data.frame` (or `tibble` if converted by the user).

See Also

[predict_quality](#)

Examples

```
# Example with package sample data
file_path <- system.file("extdata", "sample_coh.csv", package = "writeAlizer")
coh_file <- import_coh(file_path)
head(coh_file)
```

import_gamet

Import a GAMET output file into R.

Description

Import a GAMET output file into R.

Usage

```
import_gamet(path)
```

Arguments

path A string giving the path and filename to import. Keep the original CSV column names. IDs must be nonblank and unique; leading zeros are preserved. Rows are sorted by ID.

Value

A base `data.frame` with one row per record and the following columns:

- ID (character): unique identifier of the text/essay.
- One column per retained GAMET error/category variable (numeric; typically counts or rates). Column names follow the GAMET output variable names.

The object has class `data.frame` (or `tibble` if converted by the user).

See Also

[predict_quality](#)

Examples

```
# Example with package sample data
file_path <- system.file("extdata", "sample_gamet.csv", package = "writeAlizer")
gamet_file <- import_gamet(file_path)
head(gamet_file)
```

`import_merge_gamet_rb` *Import a ReaderBench output file (.csv) and GAMET output file (.csv), and merge the two files on ID.*

Description

Import a ReaderBench output file (.csv) and GAMET output file (.csv), and merge the two files on ID.

Usage

```
import_merge_gamet_rb(rb_path, gamet_path)
```

Arguments

`rb_path` A string giving the path and ReaderBench filename to import.
`gamet_path` A string giving the path and GAMET filename to import.

Value

A base `data.frame` created by joining the ReaderBench and GAMET tables by ID, with one row per matched ID and the following columns:

- ID (character): identifier present in both sources.
- All retained ReaderBench feature columns (numeric).
- All retained GAMET error/category columns (numeric).

By default, only IDs present in both inputs are kept (inner join). If a feature name appears in both sources, standard merge suffixes (e.g., `.x/.y`) may be applied by the join implementation. The object has class `data.frame` (or `tibble` if converted by the user).

See Also

[predict_quality](#)

Examples

```
# Example with package sample data
rb_path <- system.file("extdata", "sample_rb.csv", package = "writeAlizer")
gam_path <- system.file("extdata", "sample_gamet.csv", package = "writeAlizer")
rb_gam <- import_merge_gamet_rb(rb_path, gam_path)
head(rb_gam)
```

import_rb

Import a ReaderBench output file (.csv) into R.

Description

When available, the function reads the header of the packaged sample (inst/extdata/sample_rb.csv) and keeps the first 404 columns by NAME (plus the File.name/ID column), excluding any columns with names appearing after position 404 in that header. If the sample is unavailable, it falls back to keeping the first 404 columns by position.

Usage

```
import_rb(path)
```

Arguments

path	A string giving the path and filename to import. Keep the original CSV column names. IDs must be nonblank and unique; leading zeros are preserved. Rows are sorted by ID.
------	---

Value

A base data.frame with one row per record and the following columns:

- ID (character): unique identifier of the text/essay.
- One column per retained ReaderBench feature, kept by original feature name (numeric). Feature names mirror the ReaderBench output variables.

The object has class data.frame (or tibble if converted by the user).

See Also

[predict_quality](#)

Examples

```
# Fast, runnable example with package sample data
file_path <- system.file("extdata", "sample_rb.csv", package = "writeAlizer")
rb_file <- import_rb(file_path)
head(rb_file)
```

```
keep_stem_before_txt  Extract the filename stem before ".txt"
```

Description

Removes any directory path and optional ‘.txt’ extension from filenames or file paths. This function standardizes text identifiers across Coh-Metrix, GAMET, and other text analysis outputs that may include full paths or extensions in their ID fields.

Usage

```
keep_stem_before_txt(x)
```

Arguments

x A character vector (or coercible) containing file paths or filenames. Elements may or may not include a ‘.txt’ suffix or any directory path.

Details

The function handles both forward (‘/’) and backward (‘\’) slashes in file paths. If a value has no path and/or no ‘.txt’ suffix, it is returned unchanged (aside from coercion to character).

Value

A character vector where each element is reduced to the final path component, with any trailing ‘.txt’ (case-insensitive) removed. ‘NA’ values are preserved as ‘NA_character_’.

Examples

```
keep_stem_before_txt(c(
  "C:/data/3401.txt",
  "E:\\\\samples\\\\\\1002.TXT",
  "plain_id",
  NA
))
#> [1] "3401" "1002" "plain_id" NA
```

`model_deps`*Report optional model dependencies (no installation performed)*

Description

Reports optional packages from the package ‘Suggests’ field, including model, documentation, and testing packages. This function **never installs** packages. It reports which packages are required and which are currently missing, and prints a ready-to-copy command you can run to install the missing ones manually.

Usage

```
model_deps()
```

Details

You can add to the discovered packages for testing or CI with ‘options(writeAlizer.required_pkgs = c("pkgA", "pkgB (>= 1.2.3)"))’. Any version qualifiers you include are preserved in the ‘required’ output, but stripped for the availability check in ‘missing’. Installed versions are not compared against those qualifiers.

Value

A named list:

required Character vector of discovered package tokens (may include version qualifiers), e.g. `c("glmnet (>= 4.1)", "ranger")`. This is the union of the package *Suggests* field and the optional `writeAlizer.required_pkgs` override.

missing Character vector of base package names that are not installed, e.g. `c("glmnet", "ranger")`.

The function also emits a message. If nothing is missing, it reports that all required packages are installed. Otherwise, it lists the missing packages and prints a copy-paste `install.packages()` command.

Examples

```
md <- model_deps()
md$missing
```

predict_quality	<i>Predict writing quality</i>
-----------------	--------------------------------

Description

Run the specified model(s) on preprocessed data and return predictions. Apply scoring models to ReaderBench, Coh-Metrix, and/or GAMET files. Holistic writing quality can be generated from ReaderBench (model = 'rb_mod3all') or Coh-Metrix files (model = 'coh_mod3all'). Also, Total Words Written, Words Spelled Correctly, Correct Word Sequences, and Correct Minus Incorrect Word Sequences can be generated from a GAMET file (model = 'gamet_cws1').

Usage

```
predict_quality(model, data)
```

Arguments

model	A string telling which scoring model to use. ReaderBench Model 3 keys also accept a '_v2' suffix. The 'example' key is an offline demonstration. Options are: 'rb_mod1', 'rb_mod2', 'rb_mod3narr', 'rb_mod3exp', 'rb_mod3per', or 'rb_mod3all', for ReaderBench files to generate holistic quality, 'coh_mod1', 'coh_mod2', 'coh_mod3narr', 'coh_mod3exp', 'coh_mod3per', or 'coh_mod3all' for Coh-Metrix files to generate holistic quality, and 'gamet_cws1' to generate Total Words Written (TWW), Words Spelled Correctly (WSC), Correct Word Sequences (CWS) and Correct Minus Incorrect Word Sequences (CIWS) scores from a GAMET file.
data	Data frame returned by import_gamet , import_coh , or import_rb .

Details

Models 2 and 3 center and scale features using the data supplied in this call. Changing the scoring group can change a text's score. A single row or features with no variation can produce missing values. Model 1 and GAMET pass the input through to their saved models without this additional scaling.

The 'example' model is for demonstrating the workflow only. Its preprocessing needs no downloads; prediction requires `wa_seed_example_models()` first. The temporary files created for the example are cleaned up at the end of the `\examples{}`.

Value

A data.frame with ID and one column per sub-model prediction. If multiple sub-models are used and all predictions are numeric, an aggregate column named `pred_<model>_mean` is added (except for "gamet_cws1"). Missing component scores are omitted from the mean; an entirely missing row yields NaN. GAMET returns `pred_TWW_gamet`, `pred_WSC_gamet`, `pred_CWS_mod1a`, and `pred_CIWS_mod1a`. Predictions are not rounded or clipped.

See Also

[import_rb](#), [import_coh](#), [import_gamet](#)

Examples

```
local({
  old <- options(writeAlizer.mock_dir = NULL, writeAlizer.offline = TRUE)
  on.exit(options(old))
  parent <- tempfile("wa-example-")
  wa_seed_example_models(dir = parent)
  on.exit(unlink(parent, recursive = TRUE), add = TRUE)
  coh <- import_coh(system.file("extdata", "sample_coh.csv", package = "writeAlizer"))
  head(predict_quality("example", coh))
})

# Longer, networked demos
## Not run:
if (!isTRUE(getOption("writeAlizer.offline", FALSE))) {
  rb <- import_rb(system.file("extdata", "sample_rb.csv", package = "writeAlizer"))
  print(head(predict_quality("rb_mod3all", rb)))

  coh <- import_coh(system.file("extdata", "sample_coh.csv", package = "writeAlizer"))
  print(head(predict_quality("coh_mod3all", coh)))

  gam <- import_gamet(system.file("extdata", "sample_gamet.csv", package = "writeAlizer"))
  print(head(predict_quality("gamet_cws1", gam)))
}

## End(Not run)
```

preprocess	<i>Pre-process data</i>
------------	-------------------------

Description

Pre-process Coh-Metrix and ReaderBench data files before applying predictive models. Uses the artifact registry to load the correct variable lists and applies centering and scaling per sub-model, preserving the original behavior by model key.

Usage

`preprocess(model, data)`

Arguments

model	Character scalar. Which scoring model to use. Supported values include: Reader-Bench: 'rb_mod1', 'rb_mod2', 'rb_mod3narr', 'rb_mod3exp', 'rb_mod3per', 'rb_mod3all', 'rb_mod3narr_v2', 'rb_mod3exp_v2', 'rb_mod3per_v2', 'rb_mod3all_v2'; Coh-Metrix: 'coh_mod1', 'coh_mod2', 'coh_mod3narr', 'coh_mod3exp', 'coh_mod3per', 'coh_mod3all';
-------	--

	GAMET: 'gamet_cws1'. Legacy keys for RB mod3 (non-v2) are mapped to their v2 equivalents internally.
data	A data.frame produced by <code>import_rb</code> , <code>import_coh</code> , or <code>import_gamet</code> , with an ID column and the expected feature columns.

Details

Models 2 and 3 center and scale features using the data supplied in this call. Changing the scoring group can change a text's score. A single row or features with no variation can produce missing values. Model 1 and GAMET pass the input through to their saved models without this additional scaling.

The 'example' model is for demonstrating the workflow only. Its preprocessing needs no downloads; prediction requires `wa_seed_example_models()` first.

Value

A list of pre-processed data frames, one per sub-model. For models with no varlists (e.g., 'rb_mod1', 'coh_mod1'), returns six copies of the input data. For 'gamet_cws1', returns two copies (CWS/CIWS). For 1-part/3-part models, returns a list of length 1/3 with centered & scaled features plus the ID column.

Examples

```
# Minimal, offline example using the built-in 'example' model (no downloads)
rb_path <- system.file("extdata", "sample_rb.csv", package = "writeAlizer")
rb <- import_rb(rb_path)

pp <- preprocess("example", rb)
length(pp); lapply(pp, nrow)
```

wa_cache_clear	<i>Clear writeAlizer's user cache</i>
----------------	---------------------------------------

Description

Deletes all files under `wa_cache_dir()`. If `ask = TRUE` *and* in an interactive session, a short preview (item count, total size, and up to 10 sample paths) is printed before asking for confirmation. In a non-interactive script, no prompt is shown, even if `ask = TRUE`. Use this only on a dedicated cache folder; all its contents are removed.

Usage

```
wa_cache_clear(ask = interactive(), preview = TRUE)
```

Arguments

ask	Logical; if TRUE and interactive, ask for confirmation.
preview	Logical; if TRUE and ask is TRUE, show a brief listing/size summary before asking.

Value

Invisibly returns TRUE if the cache was cleared (or already absent), FALSE if the user declined or deletion failed.

See Also

[wa_cache_dir](#)

Examples

```
local({
  old <- options(writeAlizer.cache_dir = tempfile("wa-cache-"))
  on.exit(options(old))
  dir.create(wa_cache_dir())
  on.exit(unlink(wa_cache_dir(), recursive = TRUE), add = TRUE, after = FALSE)
  writelines("demo", file.path(wa_cache_dir(), "demo.txt"))
  wa_cache_clear(ask = FALSE)
})
```

wa_cache_dir

Path to writeAlizer's user cache

Description

Returns the directory used to store cached model artifacts. By default this is a platform-appropriate user cache path from `tools::R_user_dir("writeAlizer", "cache")`. If the option `writeAlizer.cache_dir` is set to a non-empty string, that location is used instead. This makes it easy to redirect the cache during tests or examples (e.g., to `tempdir()`).

Usage

```
wa_cache_dir()
```

Details

The override must be one non-missing character string. An empty string uses the default. Choose a dedicated directory: clearing the cache deletes all of its contents.

Value

Character scalar path.

See Also

[wa_cache_clear](#)

Examples

```
wa_cache_dir() # Inspect the path without creating or deleting files
```

wa_download	<i>Download and cache an artifact (graceful offline behavior)</i>
-------------	---

Description

Public helper to fetch an artifact into the user cache. This function delegates to the internal downloader used by the package at runtime, so it benefits from the same behavior:

Usage

```
wa_download(file, url, sha256 = NULL, quiet = TRUE)

download(file, url) # deprecated
```

Arguments

file	Character scalar; relative filename to use in the cache (e.g., "rb_mod1a.rda").
url	Character scalar; source URL. May be a 'file://' URL for local testing.
sha256	Optional 64-hex SHA-256 checksum for verification. If provided, the cached file must match it (or a re-download is attempted). NULL, NA, and an empty string mean no checksum. Local mock artifacts bypass checksums.
quiet	Logical; if 'TRUE', suppresses download progress messages.

Details

- Respects `options(writeAlizer.mock_dir)` to load local mock copies (useful for tests/examples and offline runs).
- Fails *gracefully* with a clear, informative message when Internet resources are unavailable or have changed (per CRAN policy).
- Verifies an optional SHA-256 checksum and re-downloads or errors if it does not match.

Local file URLs work offline. Internet URLs require a connection unless a valid cached copy exists. Downloads are verified before replacing a cached file, and failed transfers leave the previous copy intact. Nested relative filenames are accepted; absolute names and parent-directory traversal are not. The filename is the cache key: changing only the URL does not refresh an existing file. Supply the expected checksum or clear the cache to refresh it.

Value

A character scalar: the absolute path to the cached file, or the local artifact path when `writeAlizer.mock_dir` is set.

Examples

```
local({
  cache <- tempfile("wa-cache-")
  old <- options(writeAlizer.cache_dir = cache,
                 writeAlizer.mock_dir = NULL, writeAlizer.offline = TRUE)
  on.exit(options(old))
```

```

on.exit(unlink(cache, recursive = TRUE), add = TRUE)
src <- tempfile(fileext = ".bin")
writeBin(as.raw(1:10), src)
on.exit(unlink(src), add = TRUE)
path <- normalizePath(src, winslash = "/")
url <- paste0(if (.Platform$OS.type == "windows") "file:/" else "file://",
              utils::URLencode(path))
sha <- digest::digest(src, algo = "sha256", file = TRUE)
dest <- wa_download("example.bin", url, sha256 = sha)
file.exists(dest)
})

```

wa_seed_example_models

Seed example model files in a temporary directory

Description

This helper writes a minimal model file to a subdirectory of ‘dir’ (default: ‘tempdir()’), and sets the option ‘writeAlizer.mock_dir’ to that location so examples can run without downloads or network access.

Usage

```
wa_seed_example_models(model = c("example"), dir = tempdir())
```

Arguments

model	Character scalar. Only "example" is currently supported.
dir	Directory in which to create the example model (default: ‘tempdir()’).

Details

Creates an ultra-tiny model artifact used in examples and points the package loader to it via a temporary option.

- Writes under the supplied ‘dir’ (by default ‘tempdir()’) and returns the path. - The example predicts a constant 1.5; it is not a writing assessment. - Sets ‘options(writeAlizer.mock_dir = <path>)’; callers should restore prior options when appropriate (see Examples).

Value

(Invisibly) the path to the created example model directory.

Examples

```
local({
  old <- options(writeAlizer.mock_dir = NULL)
  on.exit(options(old))
  parent <- tempfile("wa-example-")
  ex <- wa_seed_example_models(dir = parent)
  on.exit(unlink(parent, recursive = TRUE), add = TRUE)
  predict_quality("example", data.frame(ID = c("text1", "text2")))
})
```

Index

`download (wa_download)`, [13](#)

`import_coh`, [3](#), [3](#), [9–11](#)

`import_gamet`, [3](#), [4](#), [9–11](#)

`import_merge_gamet_rb`, [3](#), [5](#)

`import_rb`, [3](#), [6](#), [9–11](#)

`keep_stem_before_txt`, [7](#)

`model_deps`, [3](#), [8](#)

`predict_quality`, [3–6](#), [9](#)

`preprocess`, [10](#)

`wa_cache_clear`, [3](#), [11](#), [12](#)

`wa_cache_dir`, [3](#), [12](#), [12](#)

`wa_download`, [13](#)

`wa_seed_example_models`, [14](#)

`writeAlizer (writeAlizer-package)`, [2](#)

`writeAlizer-package`, [2](#)