Package: wrictools (via r-universe)

May 7, 2026

Title Analyze Whole Room Indirect Calorimetry (WRIC) Data

Version 1.0.1

Description Provides functions, tutorials, and examples to preprocess, analyze, and visualize data from whole room indirect calorimeters (WRIC) by Maastricht Instruments, using the 'OmniCal' software. Some functions may also work with WRICs from other manufacturers, though full functionality has only been validated for Maastricht Instruments devices.

License GPL (>= 3)

Encoding UTF-8

RoxygenNote 7.3.3

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

Config/testthat/edition 3

Imports dplyr, ggplot2, magrittr, RCurl, readr, rlang, tidyr, readxl

URL https://github.com/NinaZiegenbein/wrictools, https://ninaziegenbein.github.io/wrictools/

BugReports https://github.com/NinaZiegenbein/wrictools/issues

VignetteBuilder knitr

NeedsCompilation no

Author Nina Ziegenbein [aut, cre, cph] (ORCID: <https://orcid.org/0009-0005-4913-8910>), Anders Isaksen [rev] (ORCID: <https://orcid.org/0000-0001-8457-5466>)

Maintainer Nina Ziegenbein <nina.ziegenbein@web.de>

Repository https://cran.r-universe.dev

Date/Publication 2026-05-07 22:31:49 UTC

RemoteUrl https://github.com/cran/wrictools

RemoteRef HEAD

RemoteSha 6813da8a2923a3c7d6cd94af335562cd434def05

Contents

`add_relative_time` *Add Relative Time in minutes to DataFrame. Rows before the* `start_time` *will be indicated negative.*

Description

Add Relative Time in minutes to DataFrame. Rows before the `start_time` will be indicated negative.

Usage

```
add_relative_time(df, start_time = NULL)
```

Arguments

`df`	A data frame containing a 'datetime' column.
`start_time`	Optional; the starting time for calculating relative time. Should be in a format compatible with POSIXct (eg. "2023-11-13 11:40:00")

Value

A data frame with an additional column 'relative_timemin' indicating the time in minutes from the start time.

Examples

```
# Create example data
df <- data.frame(
 datetime = as.POSIXct(c("2023-11-13 11:40:00", "2023-11-13 11:45:00", "2023-11-13 11:50:00"))
)
add_relative_time(df)
add_relative_time(df, start_time = "2023-11-13 11:45:00")
```

`analyse_methanol_burn` *Analyse methanol burn experiment*

Description

Analyse methanol burn experiment

Usage

```
analyse_methanol_burn(
  filepath,
  methanolfilepath,
  room1 = TRUE,
  datetime_format = "%d-%m-%y %H:%M",
  code = "id",
  manual = NULL,
  save_csv = FALSE,
  path_to_save = NULL,
  combine = TRUE,
  method = "mean",
  start = NULL,
  end = NULL,
  notefilepath = NULL,
  keywords_dict = NULL,
  entry_exit_dict = NULL
)
```

Arguments

`filepath`	Path to the wric .txt file.
`methanolfilepath`	File path to a csv or Excel file with columns: **datetime** Timestamp of measurement, format `%d-%m-%y %H:%M` by default. Can override with `datetime_format`. **methanol** Methanol mass (g) at each timestamp.
`room1`	Logical; if TRUE uses room1 data from WRIC file, else room2. This is only relevant for files generates by software version 1. Default is TRUE.

`datetime_format`
: Character; format string for parsing methanol datetime column. Default is "%d-%m-%y %H:%M" (25-03-25 13:58).

`code`
: Method for generating subject IDs ("id", "id+comment", "study+id" (only for software v2), or "manual").

`manual`
: Custom codes for subjects in Room 1 and Room 2 if `code` is "manual".

`save_csv`
: Logical, whether to save extracted metadata and data to CSV files.

`path_to_save`
: Directory path for saving CSV files, NULL uses the current directory.

`combine`
: Logical, whether to combine S1 and S2 measurements.

`method`
: Method for combining measurements ("mean", "median", "s1", "s2", "min", "max").

`start`
: character or POSIXct or NULL, rows before this will be removed, if NULL takes first row e.g "2023-11-13 11:43:00"

`end`
: character or POSIXct or NULL, rows after this will be removed, if NULL takes last row e.g "2023-11-13 11:43:00"

`notefilepath`
: String, Directory path of the corresponding note file (.txt)

`keywords_dict`
: List, A dictionary of keywords used to extract protocol events from a note file. Each entry should be a named list with:

  keywords A character vector of keywords or phrases to match in the note comment. Matching is case-insensitive.

  value Numeric protocol value to assign when the keyword is detected.

  type Optional character, either "instant" or omitted. "instant" events are applied at the specified timestamp and revert to the previous protocol immediately after. Non-instant events set the protocol until another event occurs.

  Behavior rules:

  Non-instant events Set the protocol value from their timestamp until another event overwrites it or until a stop keyword sets it to 0.

  Instant events Apply only at the timestamp of the note line, then revert to the protocol that was active immediately before.

  Stop keywords Always set the protocol to 0, regardless of previous state, unless overridden by an instant event.

  If `NULL`, a default set of keywords is used.

`entry_exit_dict`
: Nested List, used to extract entry/exit times from note file

Value

A list with:

per_interval Data frame with per-timestep methanol burn, predicted CO2/O2, measured VO2/VCO2, deviations, RER

overall Summary for the whole session

plots List of ggplot objects (values shown correspond to the end of each interval)

Examples

```
methanol <- source(path.expand("~/methanol.xlsx"))
data_txt <- system.file("extdata", "data.txt", package = "wrictools")
analyse_methanol_burn (data_txt, methanol, room1 = FALSE)
```

`analyse_zero_test` *Analyse a Zero Test WRIC file*

Description

This function preprocesses a WRIC data file, plots VO2 and VCO2 over time, and returns basic statistics (mean, SD, min, max, slope). For version 1 files, separate plots and statistics are returned for Room 1 and Room 2. For version 2 files, a single dataset is processed.

Usage

```
analyse_zero_test(
  filepath,
  code = "id",
  manual = NULL,
  save_csv = FALSE,
  path_to_save = NULL,
  combine = TRUE,
  method = "mean",
  start = NULL,
  end = NULL,
  notefilepath = NULL,
  keywords_dict = NULL,
  entry_exit_dict = NULL,
  verbose = TRUE
)
```

Arguments

`filepath`	Path to the wric .txt file.
`code`	Method for generating subject IDs ("id", "id+comment", "study+id" (only for software v2), or "manual").
`manual`	Custom codes for subjects in Room 1 and Room 2 if code is "manual".
`save_csv`	Logical, whether to save extracted metadata and data to CSV files.
`path_to_save`	Directory path for saving CSV files, NULL uses the current directory.
`combine`	Logical, whether to combine S1 and S2 measurements.
`method`	Method for combining measurements ("mean", "median", "s1", "s2", "min", "max").

`start` character or POSIXct or NULL, rows before this will be removed, if NULL takes first row e.g "2023-11-13 11:43:00"

`end` character or POSIXct or NULL, rows after this will be removed, if NULL takes last row e.g "2023-11-13 11:43:00"

`notefilepath` String, Directory path of the corresponding note file (.txt)

`keywords_dict` List, A dictionary of keywords used to extract protocol events from a note file. Each entry should be a named list with:

keywords A character vector of keywords or phrases to match in the note comment. Matching is case-insensitive.

value Numeric protocol value to assign when the keyword is detected.

type Optional character, either "instant" or omitted. "instant" events are applied at the specified timestamp and revert to the previous protocol immediately after. Non-instant events set the protocol until another event occurs.

Behavior rules:

Non-instant events Set the protocol value from their timestamp until another event overwrites it or until a stop keyword sets it to 0.

Instant events Apply only at the timestamp of the note line, then revert to the protocol that was active immediately before.

Stop keywords Always set the protocol to 0, regardless of previous state, unless overridden by an instant event.

If `NULL`, a default set of keywords is used.

`entry_exit_dict`

Nested List, used to extract entry/exit times from note file

`verbose` Logical; if TRUE (default), prints the plot and statistics.

Value

A named list of data frames with statistics for each room (v1) or for all data (v2).

Examples

```
filepath <- system.file("extdata", "data.txt", package = "wrictools")
analyse_zero_test(filepath)
```

---

check_code *Check the subject ID code and return corresponding Room 1 and Room 2 codes.*

Description

Check the subject ID code and return corresponding Room 1 and Room 2 codes.

Usage

```
check_code(code, manual, metadata, v1 = FALSE)
```

Arguments

code	Method for generating subject IDs ("id", "id+comment", "study+id", or "manual").
manual	A custom code(string), required if code is "manual".
metadata	DataFrame for metadata of Room 1.
v1	Boolean, Software Version, default FALSE.

Value

String, the resulting code.

Examples

```
# Example metadata
metadata <- data.frame(`Subject.ID` = "S001", `Study.ID` = "studyname", `Comments` = "Morning")

# Use subject ID only
check_code("id", NULL, metadata)

# Use subject ID + comment
check_code("id+comment", NULL, metadata)

# Use study ID + subject ID
check_code("study+id", NULL, metadata)

# Use manual codes
check_code("manual", "custom1", metadata)
```

---

check_discrepancies *Checks for discrepancies between S1 and S2 measurements in the DataFrame and prints them to the console. This function is not included in the big pre-processing function, as it is more intended to perform a quality check on your data and not to automatically inform the processing of the data.*

Description

Checks for discrepancies between S1 and S2 measurements in the DataFrame and prints them to the console. This function is not included in the big pre-processing function, as it is more intended to perform a quality check on your data and not to automatically inform the processing of the data.

Usage

```
check_discrepancies(df, threshold = 0.05, individual = FALSE)
```

Arguments

df	DataFrame containing wric data with columns for S1 and S2 measurements.
threshold	Numeric threshold percentage for mean relative delta discrepancies (default 0.05).
individual	Logical, if TRUE checks and reports individual row discrepancies beyond the threshold (default FALSE).

Value

Returns the discrepancies as a single character vector.

Examples

```
data_txt <- system.file("extdata", "data.txt", package = "wrictools")
lines <- readLines(data_txt)
# Create example WRIC data frames
result <- create_wric_df(
  filepath = data_txt,
  lines = lines,
  code_1 = "R1",
  code_2 = "R2",
  path_to_save = tempdir(),
  start = NULL,
  end = NULL,
  notefilepath = NULL
)
check_discrepancies(result$df_room1)
```

---

combine_measurements *Combines S1 and S2 measurements in the DataFrame using the specified method.*

Description

Combines S1 and S2 measurements in the DataFrame using the specified method.

Usage

```
combine_measurements(df, method = "mean")
```

Arguments

df	DataFrame containing wric data with S1 and S2 measurement columns.
method	String specifying the method to combine measurements ("mean", "median", "s1", "s2", "min", "max").

Value

A DataFrame with combined measurements.

Examples

```
data_txt <- system.file("extdata", "data.txt", package = "wrictools")
lines <- readLines(data_txt)

# Create example WRIC DataFrames
result <- create_wric_df(
  filepath = data_txt,
  lines = lines,
  code_1 = "R1",
  code_2 = "R2",
  path_to_save = tempdir(),
  start = NULL,
  end = NULL,
  notefilepath = NULL
)

# Combine measurements using different methods
combined_mean <- combine_measurements(result$df_room1, method = "mean")
combined_median <- combine_measurements(result$df_room1, method = "median")
combined_s1 <- combine_measurements(result$df_room1)
```

`create_wric_df` *Creates DataFrames for wric data from a file and optionally saves them as CSV files.*

Description

Creates DataFrames for wric data from a file and optionally saves them as CSV files.

Usage

```
create_wric_df(
  filepath,
  lines,
  code_1,
  code_2,
  path_to_save = NULL,
  start = NULL,
  end = NULL,
  notefilepath = NULL,
  entry_exit_dict = NULL
)
```

Arguments

`filepath`	Path to the wric .txt file.
`lines`	List of strings read from the file to locate the data start.

code_1	String representing the codes for Room 1.
code_2	String representing the codes for Room 2.
path_to_save	Directory path for saving CSV files, NULL uses the current directory.
start	character or POSIXct or NULL, rows before this will be removed, if NULL takes first row e.g "2023-11-13 11:43:00"
end	character or POSIXct or NULL, rows after this will be removed, if NULL takes last row e.g "2023-11-13 11:43:00"
notefilepath	String, The path to the notefile
entry_exit_dict	Nested List, used to extract entry/exit times from note file

Value

A list containing DataFrames for Room 1 and Room 2 measurements.

Note

Raises an error if Date or Time columns are inconsistent across rows.

Examples

```
# Load example files from the package
data_txt <- system.file("extdata", "data.txt", package = "wrictools")
notes_txt <- system.file("extdata", "note.txt", package = "wrictools")

# Create the data lines for parsing
lines <- readLines(data_txt)

# Call the function
result <- create_wric_df(
  filepath = data_txt,
  lines = lines,
  code_1 = "XXXX",
  code_2 = "YYYY",
  path_to_save = tempdir(),
  start = NULL,
  end = NULL,
  notefilepath = notes_txt
)
```

---

create_wric_df_new *Creates a DataFrame for WRIC data from the new Omnical software format.*

Description

Creates a DataFrame for WRIC data from the new Omnical software format.

Usage

```
create_wric_df_new(
  filepath,
  lines,
  code,
  path_to_save = NULL,
  start = NULL,
  end = NULL,
  notefilepath = NULL,
  entry_exit_dict = NULL
)
```

Arguments

filepath	Path to the new-format WRIC .txt file.
lines	List of strings read from the file to locate the data start (used to find "Set 1").
code	String representing the study or participant code, used for naming outputs.
path_to_save	Directory path for saving CSV files or outputs. Currently not used for saving; default NULL.
start	Character or POSIXct or NULL. Rows before this time will be removed. If NULL, uses the earliest available row.
end	Character or POSIXct or NULL. Rows after this time will be removed. If NULL, uses the latest available row.
notefilepath	String or NULL. Path to a note file. If provided, `detect_start_end()` is called to determine start and end times.
entry_exit_dict	Nested list, used by `detect_start_end()` to extract entry/exit times from note file.

Value

A data frame containing the parsed WRIC measurements, including all sets (S1 and S2), a `datetime` column (POSIXct), and `relative_time` column (seconds from start).

Note

- Raises an error if the "Set 1" header cannot be found in the file.
- Raises an error if Date or Time columns are inconsistent across sets in any row.
- Handles the extra empty column between Set 1 and Set 2 to avoid parsing issues.

Examples

```
# Load example files from the package
data_v2_txt <- system.file("extdata", "data_v2.txt", package = "wrictools")
notes_v2_txt <- system.file("extdata", "note_v2.txt", package = "wrictools")

# Create the data lines for parsing
```

```
lines <- readLines(data_v2_txt)

# Call the function
df <- create_wric_df_new(
  filepath = data_v2_txt,
  lines = lines,
  code = "study+id",
  path_to_save = NULL,
  start = NULL,
  end = NULL,
  notefilepath = notes_v2_txt
)
```

`cut_rows` *Filters rows in a DataFrame based on an optional start and end datetime range.*

Description

Filters rows in a DataFrame based on an optional start and end datetime range.

Usage

```
cut_rows(df, start = NULL, end = NULL)
```

Arguments

`df`	data.frame DataFrame with a "datetime" column to filter.
`start`	character or POSIXct or NULL, optional; Start datetime; rows before this will be removed. If NULL, uses the earliest datetime in the DataFrame.
`end`	character or POSIXct or NULL, optional End datetime; rows after this will be removed. If NULL, uses the latest datetime in the DataFrame.

Details

Throws an error if filtering by start and end results in an empty DataFrame: no rows remain after applying the start/end window.

Value

data.frame DataFrame with rows between the specified start and end dates, or the full DataFrame if both are NULL.

Examples

```
df <- data.frame(
  datetime = as.POSIXct(c(
    "2023-11-13 11:40:00",
    "2023-11-13 11:45:00",
    "2023-11-13 11:50:00"
  ))
)

# Filter rows from 11:45 onward
cut_rows(df, start = "2023-11-13 11:45:00")

# Filter rows between 11:40 and 11:45
cut_rows(df, start = "2023-11-13 11:40:00", end = "2023-11-13 11:45:00")

# No filtering (both NULL)
cut_rows(df)
```

`detect_start_end` *Automatically detect enter and exit from the chamber based on the notefile. Returns the start and end times for two participants.*

Description

Automatically detect enter and exit from the chamber based on the notefile. Returns the start and end times for two participants.

Usage

```
detect_start_end(notes_path, v1 = FALSE, entry_exit_dict = NULL)
```

Arguments

`notes_path`	string - path to the note file
`v1`	Boolean, Software Version, default FALSE.
`entry_exit_dict`	Nested List, used to extract entry/exit times from note file

Value

list - A list of two elements ("1" and "2"), each containing a tuple (start, end) time. Returns NA if not possible to find start or end time.

Examples

```
notes_path <- system.file("extdata", "note.txt", package = "wrictools")
detect_start_end(notes_path)
```

`export_file_from_redcap` *Exports a file from REDCap based on the specified record ID and field name.*

Description

If you do not specify a path, the data will be downloaded to a temporary folder which is deleted when your R session ends.

Usage

```
export_file_from_redcap(record_id, fieldname, path = NULL, api_url, api_token)
```

Arguments

`record_id`	String containing the unique identifier for the record in REDCap.
`fieldname`	Field name from which to export the file.
`path`	File path where the exported file will be saved.
`api_url`	String, URL to the REDCap API, should be specified in your personal config.R file
`api_token`	String, personal token for the REDCap API, should be specified in your personal config.R file

Value

filepath String, the filepath that the file was just saved to.

Examples

```
source(path.expand("~/.config.R"))
export_file_from_redcap(
  record_id = "1", fieldname = "wric_data",
  api_url = api_url, api_token = api_token
)
```

`extract_meta_data` *Extracts metadata (software v1) for two subjects from text lines and optionally saves it as CSV files.*

Description

Extracts metadata (software v1) for two subjects from text lines and optionally saves it as CSV files.

Usage

```
extract_meta_data(lines, code, manual, save_csv = FALSE, path_to_save)
```

Arguments

`lines`	List of strings containing the wric metadata.
`code`	Method for generating subject IDs ("id", "id+comment", "study+id" (only for software v2), or "manual").
`manual`	Custom codes for Room 1 and Room 2 subjects if `code` is "manual".
`save_csv`	Logical, whether to save extracted metadata to CSV files.
`path_to_save`	Directory path for saving CSV files, NULL uses the current directory.

Value

A list containing the Room 1 code, Room 2 code, and DataFrames for r1_metadata and r2_metadata.

Examples

```
lines <- c(
  "OmniCal software by ing.P.F.M.Schoffelen, Dept. of Human Biology, Maastricht University",
  "file identifier is C:\\MI_Room_Calorimeter\\Results_online\\1_minute\\Results.txt",
  "",
  "Room 1\tProject\tSubject ID\tExperiment performed by\tComments",
  "\tPROJECT\tXXXX\tJANE DOE\t",
  "Room 2\tProject\tSubject ID\tExperiment performed by\tComments",
  "\tPROJECT\tYYYY\tJOHN DOE\t"
)

extract_meta_data(lines, code = "id", manual = NULL, save_csv = FALSE, path_to_save = NULL)
```

`extract_metadata_new` *Extracts metadata (software v2) for a single subject from text lines and optionally saves it as a CSV file.*

Description

Extracts metadata (software v2) for a single subject from text lines and optionally saves it as a CSV file.

Usage

```
extract_metadata_new(
  lines,
  code,
  manual = NULL,
  save_csv = FALSE,
  path_to_save = NULL
)
```

Arguments

`lines`	List of strings containing the wric metadata.
`code`	Method for generating subject IDs ("id", "id+comment", "study+id" (only for software v2), or "manual").
`manual`	Custom code for the subject if `code` is "manual".
`save_csv`	Logical, whether to save extracted metadata to a CSV file.
`path_to_save`	Directory path for saving the CSV file, NULL uses the current directory.

Value

A list containing:

code The generated subject code.

metadata A data.frame containing the extracted metadata.

Examples

```
lines <- c(
  "2.0.0.15\tOmnical software by Maastricht Instruments B.V.",
  "file identifier is C:\\Omnical\\Results Room 1\\1 min\\Results.txt",
  "Calibration values\t18.0\t0.795",
  "",
  "Subject ID\tStudy ID\tMeasurement ID\tResearcher ID\tComments",
  "ZeroTest\t\t\tNZ\tZero Test 14.01.2026"
)

# Extract metadata and generate code from subject ID only
```

```
extract_metadata_new(lines, code = "id", manual = NULL, save_csv = FALSE)

# Extract metadata using ID + comment as code
extract_metadata_new(lines, code = "id+comment", manual = NULL, save_csv = FALSE)

# Extract metadata using a manual code
extract_metadata_new(lines, code = "manual", manual = "custom_code", save_csv = FALSE)
```

`extract_note_info` *Apply protocol events from note files to room data*

Description

Reads a note file, extracts protocol events for each participant, applies any detected time drift, and updates the protocol column in the provided room data frames.

Usage

```
extract_note_info(notes_path, df_room1, df_room2, keywords_dict = NULL)
```

Arguments

`notes_path`	character Path to the note file containing protocol events.
`df_room1`	data.frame Data frame for room 1 containing at least a "datetime" column.
`df_room2`	data.frame Data frame for room 2 containing at least a "datetime" column.
`keywords_dict`	List, A dictionary of keywords used to extract protocol events from a note file. Each entry should be a named list with:

keywords A character vector of keywords or phrases to match in the note comment. Matching is case-insensitive.

value Numeric protocol value to assign when the keyword is detected.

type Optional character, either "instant" or omitted. "instant" events are applied at the specified timestamp and revert to the previous protocol immediately after. Non-instant events set the protocol until another event occurs.

Behavior rules:

Non-instant events Set the protocol value from their timestamp until another event overwrites it or until a stop keyword sets it to 0.

Instant events Apply only at the timestamp of the note line, then revert to the protocol that was active immediately before.

Stop keywords Always set the protocol to 0, regardless of previous state, unless overridden by an instant event.

If `NULL`, a default set of keywords is used.

Value

A list with two elements:

df_room1 Data frame for room 1 with updated protocol column.

df_room2 Data frame for room 2 with updated protocol column.

Examples

```
df1 <- data.frame(datetime = as.POSIXct(c("2023-11-13 22:40:00", "2023-11-13 22:50:00")))
df2 <- data.frame(datetime = as.POSIXct(c("2023-11-13 22:40:00", "2023-11-13 22:50:00")))
note_file <- system.file("extdata", "note.txt", package = "wrictools")
res <- extract_note_info(note_file, df1, df2)
res$df_room1
res$df_room2
```

---

extract_note_info_new *Apply protocol events from note files to a single room data frame (new software)*

Description

Reads a note file, extracts protocol events, applies any detected time drift, and updates the protocol column in the provided data frame. Designed for notes generated by the newer software version where all participants are in a single data frame.

Usage

```
extract_note_info_new(df, notes_path, keywords_dict = NULL)
```

Arguments

df	data.frame Data frame containing at least a "datetime" column.
notes_path	character Path to the note file containing protocol events.
keywords_dict	List, A dictionary of keywords used to extract protocol events from a note file. Each entry should be a named list with:

keywords A character vector of keywords or phrases to match in the note comment. Matching is case-insensitive.

value Numeric protocol value to assign when the keyword is detected.

type Optional character, either "instant" or omitted. "instant" events are applied at the specified timestamp and revert to the previous protocol immediately after. Non-instant events set the protocol until another event occurs.

Behavior rules:

Non-instant events Set the protocol value from their timestamp until another event overwrites it or until a stop keyword sets it to 0.

Instant events Apply only at the timestamp of the note line, then revert to the protocol that was active immediately before.

Stop keywords Always set the protocol to 0, regardless of previous state, unless overridden by an instant event.

If NULL, a default set of keywords is used.

Value

data.frame The input data frame with an updated "protocol" column based on extracted events.

Examples

```
df <- data.frame(datetime = as.POSIXct(c("2023-11-13 22:40:00", "2023-11-13 22:50:00")))
note_file <- system.file("extdata", "note_v2.txt", package = "wrictools")
df_updated <- extract_note_info_new(df, note_file)
df_updated
```

---

preprocess_wric_file *Preprocesses a wric data file, extracting metadata, creating DataFrames, and optionally saving results.*

Description

Preprocesses a wric data file, extracting metadata, creating DataFrames, and optionally saving results.

Usage

```
preprocess_wric_file(
  filepath,
  code = "id",
  manual = NULL,
  save_csv = FALSE,
  path_to_save = NULL,
  combine = TRUE,
  method = "mean",
  start = NULL,
  end = NULL,
  notefilepath = NULL,
  keywords_dict = NULL,
  entry_exit_dict = NULL
)
```

Arguments

filepath	Path to the wric .txt file.
code	Method for generating subject IDs ("id", "id+comment", "study+id" (only for software v2), or "manual").
manual	Custom codes for subjects in Room 1 and Room 2 if code is "manual".

`save_csv` Logical, whether to save extracted metadata and data to CSV files.

`path_to_save` Directory path for saving CSV files, NULL uses the current directory.

`combine` Logical, whether to combine S1 and S2 measurements.

`method` Method for combining measurements ("mean", "median", "s1", "s2", "min", "max").

`start` character or POSIXct or NULL, rows before this will be removed, if NULL takes first row e.g "2023-11-13 11:43:00"

`end` character or POSIXct or NULL, rows after this will be removed, if NULL takes last row e.g "2023-11-13 11:43:00"

`notefilepath` String, Directory path of the corresponding note file (.txt)

`keywords_dict` List, A dictionary of keywords used to extract protocol events from a note file. Each entry should be a named list with:

keywords A character vector of keywords or phrases to match in the note comment. Matching is case-insensitive.

value Numeric protocol value to assign when the keyword is detected.

type Optional character, either "instant" or omitted. "instant" events are applied at the specified timestamp and revert to the previous protocol immediately after. Non-instant events set the protocol until another event occurs.

Behavior rules:

Non-instant events Set the protocol value from their timestamp until another event overwrites it or until a stop keyword sets it to 0.

Instant events Apply only at the timestamp of the note line, then revert to the protocol that was active immediately before.

Stop keywords Always set the protocol to 0, regardless of previous state, unless overridden by an instant event.

If `NULL`, a default set of keywords is used.

`entry_exit_dict`

Nested List, used to extract entry/exit times from note file

Value

list A list with the following components:

version Character string indicating the detected software version (`"1"` for old software, `"2"` for new software).

metadata A named list containing extracted metadata. For version 1, this includes `r1` and `r2`. For version 2, this contains a single `metadata` entry.

dfs A named list containing processed data frames. For version 1: `room1` and `room2`. For version 2: `data`.

Examples

```
outdir <- file.path(tempdir(), "wrictools")
dir.create(outdir, showWarnings = FALSE)
data_txt <- system.file("extdata", "data.txt", package = "wrictools")
result <- preprocess_wric_file(data_txt, path_to_save = outdir)
unlink(outdir, recursive = TRUE)
```

preprocess_wric_files *Preprocesses multiple wric_files by RedCAP record ID, extracting metadata, creating DataFrames, and optionally saving results.*

Description

Preprocesses multiple wric_files by RedCAP record ID, extracting metadata, creating DataFrames, and optionally saving results.

Usage

```
preprocess_wric_files(
  csv_file,
  fieldname,
  code = "id",
  manual = NULL,
  save_csv = FALSE,
  path_to_save = NULL,
  combine = TRUE,
  method = "mean",
  start = NULL,
  end = NULL,
  path = NULL,
  api_url,
  api_token
)
```

Arguments

csv_file	Path to the CSV file containing record IDs.
fieldname	The field name for exporting wric data from RedCAP.
code	Method for generating subject IDs ("id", "id+comment", "study+id" (only for software v2), or "manual").
manual	Custom codes for subjects in Room 1 and Room 2 if code is "manual".
save_csv	Logical, whether to save extracted metadata and data to CSV files.
path_to_save	Directory path for saving CSV files, NULL uses the current directory.
combine	Logical, whether to combine S1 and S2 measurements.
method	Method for combining measurements ("mean", "median", "s1", "s2", "min", "max").
start	character or POSIXct or NULL, rows before this will be removed, if NULL takes first row e.g "2023-11-13 11:43:00"
end	character or POSIXct or NULL, rows after this will be removed, if NULL takes last row e.g "2023-11-13 11:43:00"
path	File path where the exported file will be saved.

api_url	String, URL to the REDCap API, should be specified in your personal config.R file
api_token	String, personal token for the REDCap API, should be specified in your personal config.R file

Value

A named list where each name corresponds to a record ID. Each element of the list is itself a list containing:

version Character, either "1" or "2" depending on the WRIC file version.

metadata List of metadata.

- **v1** List with r1 and r2 metadata for version 1 files.
- **v2** List with metadata for version 2 files.

dfs List of data frames.

- **v1** List with room1 and room2 data frames for version 1 files.
- **v2** List with data for version 2 files.

Examples

```
source(path.expand("~/.config.R"))
tmp_csv <- tempfile(fileext = ".csv")
write.csv(data.frame(X1 = c(1, 2, 3)), tmp_csv, row.names = FALSE)

# Use dummy API URL and token
if (file.exists(tmp_csv)) {
  preprocess_wric_files(
    csv_file = tmp_csv,
    fieldname = "wric_data",
    api_url = api_url,
    api_token = api_token,
    save_csv = FALSE
  )
}
```

---

upload_file_to_redcap *Uploads a file to REDCap for a specified record ID and field name.*

Description

Uploads a file to REDCap for a specified record ID and field name.

Usage

```
upload_file_to_redcap(filepath, record_id, fieldname, api_url, api_token)
```

Arguments

filepath	Path to the file to be uploaded.
record_id	String containing the unique identifier for the record in REDCap.
fieldname	Field name to which the file will be uploaded.
api_url	String, URL to the REDCap API, should be specified in your personal config.R file
api_token	String, personal token for the REDCap API, should be specified in your personal config.R file

Value

The HTTP status code of the request.

Examples

```
source(path.expand("~/.config.R"))
tmp <- tempfile(fileext = ".txt")
writeLines(c("Example content"), tmp)
upload_file_to_redcap(
  filepath = tmp, record_id = "1", fieldname = "wric_data",
  api_url = api_url, api_token = api_token
)
```

---

visualize_with_protocol

Visualizes time-series data from a WRIC CSV file, highlighting protocol changes and optionally saving the plot.

Description

Visualizes time-series data from a WRIC CSV file, highlighting protocol changes and optionally saving the plot.

Usage

```
visualize_with_protocol(
  csv_file,
  plot = "RER",
  protocol_colors_labels = NULL,
  save_png = FALSE,
  path_to_save = NULL
)
```

Arguments

csv_file	Path to the CSV file containing time-series data.
plot	A string specifying the column to plot. Defaults to "RER". This can be any valid column name in the CSV file.
protocol_colors_labels	A data frame containing the protocol codes, colors, and labels. If NULL, defaults to a predefined set of protocols.
save_png	Logical, whether to save the plot as a PNG file. Defaults to FALSE.
path_to_save	Directory path for saving the PNG file. If NULL, saves in the current working directory.

Value

A ggplot2 object visualizing the specified data with protocol highlights. Optionally saves the plot to a file if save_png is TRUE.

Examples

```
csv <- system.file("extdata", "example.csv", package = "wrictools")
figure <- visualize_with_protocol(csv, plot = "VO2")
```

Index