# Package 'actiread'

October 26, 2026

**Type** Package

**Title** Baseline Package for Reading Actigraphy and Activity Data

**Version** 0.9.0

**Description** Provides baseline functions for reading actigraphy and activity data, relying on baseline functions from 'actibase'. Reads data from 'Axivity' 'CWA' <https://axivity.com/>, ActiGraph 'GT3X' <https://academic.ametris.com/actigraph-wgt3x-bt>, 'GENEActiv' and 'Genea' binary files, 'Parmay Tech Matrix' binary files, 'SensorLog' <https://sensorlog.berndthomas.net/>, and 'SensorLogger' <https://www.tszheichoi.com/sensorlogger> zipped CSV files.

**License** GPL-3

**Depends** R (>= 4.1.0)

**Suggests** testthat (>= 3.0.0), covr, knitr, rmarkdown, reticulate, arrow

**Encoding** UTF-8

**VignetteBuilder** knitr

**Imports** actibase, GGIRread, read.gt3x (>= 0.2.0), R.utils, magrittr, dplyr, lubridate, assertthat, cli, tibble, janitor, purrr, readr, tools, utils

**URL** https://github.com/jhuwit/actiread, https://jhuwit.github.io/actiread/

**BugReports** https://github.com/jhuwit/actiread/issues

**Config/roxygen2/version** 8.1.1

**Config/testthat/edition** 3

**NeedsCompilation** no

**Author** John Muschelli [aut, cre] (ORCID: <https://orcid.org/0000-0001-6469-1750>)

**Maintainer** John Muschelli <muschellij2@gmail.com>

**Repository** CRAN

**Date/Publication** 2026-10-26 16:20:02 UTC

# Contents

acti_convert_sensorlogger_time . . . 2
acti_decompress_file . . . 3
acti_example_gt3x . . . 3
acti_py_read_cwa . . . 4
acti_read . . . 5
acti_read_cwa . . . 6
acti_read_genea . . . 7
acti_read_geneactiv . . . 8
acti_read_gt3x . . . 9
acti_read_parmay_matrix . . . 10
acti_read_sensorlog . . . 11
acti_require_actipy . . . 12
acti_rewrite_sensorlog_csv . . . 13
acti_sensorlogger_location_colnames_mapping . . . 13
tzoffset_to_tz . . . 14

**Index** **15**

---

`acti_convert_sensorlogger_time` *Convert vectors ensuring no new NA*

---

## Description

Convert vectors ensuring no new NA

## Usage

```
acti_convert_sensorlogger_time(x)
```

## Arguments

| | |
|---|---|
| x | a vector |

## Value

A converted `vector` the same length as `x` or errors if there are introduced NAs.

## acti_decompress_file *Decompress an Activity-data File*

### Description

Decompresses `.bz2`, `.gz`, and `.xz` files to a temporary file. If `extension` is `NULL`, the destination extension is inferred after removing the compression extension from `path`.

### Usage

```
acti_decompress_file(path, extension = NULL)
```

### Arguments

| | |
|---|---|
| `path` | Path to an activity-data file, optionally compressed. |
| `extension` | Output file extension. Defaults to `NULL`, which infers the extension from `path` after removing its compression extension. |

### Value

A list with `path`, the usable file path, and `temporary`, which is `TRUE` when a temporary decompressed file was created. Callers should delete `path` when `temporary` is `TRUE`.

### Examples

```
file = acti_decompress_file(acti_example_cwa())
if (file$temporary) {
  unlink(file$path)
}
```

## acti_example_gt3x *Activity/Actigraphy Example Data*

### Description

Activity/Actigraphy Example Data

### Usage

```
acti_example_gt3x()

acti_example_cwa()

acti_example_geneactiv()

acti_example_genea()
```

```
acti_example_parmay_matrix()

acti_example_sensorlog_file()

acti_example_sensorlogger_file()

acti_example_sensorlogger_location_file()
```

## Value

A file path

## Examples

```
library(actiread)
acti_example_gt3x()
acti_example_cwa()
acti_example_sensorlogger_file()
acti_example_genea()
acti_example_parmay_matrix()
```

---

# acti_py_read_cwa *Read CWA File with* actipy

---

## Description

Read CWA File with actipy

## Usage

```
acti_py_read_cwa(
  path,
  lowpass_hz = FALSE,
  calibrate_gravity = FALSE,
  detect_nonwear = FALSE,
  resample_hz = FALSE,
  ...,
  tz = "UTC",
  apply_tz = TRUE,
  verbose = TRUE
)
```

## Arguments

| | |
|---|---|
| path | Path to cwa file |
| lowpass_hz | Frequency of low pass filter |

| | |
|---|---|
| calibrate_gravity | perform gravity calibration method of van Hees et al. 2014 doi:10.1152/japplphysiol.00421.2014 |
| detect_nonwear | Flag nonwear episodes in the data by setting them to NA. Non-wear episodes are inferred from long periods of no movement. 90 minute windows based on 10-second rolling windows with a threshold of 0.015g standard deviation tolerance |
| resample_hz | Target frequency (Hz) to resample the signal. If "uniform", uses the device's sample rate to fix sampling errors. Pass FALSE to disable. |
| ... | additional arguments to pass to actipy.read_device |
| tz | time zone for the data time, passed to desiredtz argument in GGIRread::readAxivity. If NULL or "", no time conversion is done. |
| apply_tz | turn the time column into a POSIXct and apply the timezone |
| verbose | print diagnostic messages, higher values = more verbosity. |

## Value

A tibble with attributes of a header, sample rate, and transformations

## Examples

```
if (requireNamespace("reticulate", quietly = TRUE) &&
    requireNamespace("arrow", quietly = TRUE)) {
    if (reticulate::py_module_available("actipy") &&
      reticulate::py_module_available("pyarrow")) {
        data = acti_py_read_cwa(acti_example_cwa())
    }
}
```

---

# acti_read *Read an actigraphy or sensor file*

---

## Description

Guess the appropriate reader from the file extension and pass additional arguments through to that reader. Compressed .gz, .bz2, and .xz files are dispatched by their underlying extension. Because 'GENEActiv', 'Genea', and 'Parmay Matrix' devices all write .bin files, the first kilobyte of .bin files (and files with an unknown extension) is inspected to pick the reader.

## Usage

```
acti_read(path, ..., verbose = TRUE)
```

## Arguments

| | |
|---|---|
| path | Path to a file to read. |
| ... | Arguments passed to the selected reader. |
| verbose | Print the selected reader before reading. Defaults to TRUE. |

## Value

The result from the selected reader.

## Examples

```
data = acti_read(acti_example_cwa(), verbose = FALSE)
data = acti_read(acti_example_genea(), verbose = FALSE)
```

---

`acti_read_cwa` *Read CWA File*

---

## Description

Read CWA File

## Usage

```
acti_read_cwa(
  path,
  start = 0,
  end = Inf,
  tz = "UTC",
  ...,
  apply_tz = TRUE,
  verbose = TRUE
)

acti_read_cwa_header(path)
```

## Arguments

| | |
|---|---|
| `path` | Path to cwa file |
| `start` | where to start in the file, passed to GGIRread::readAxivity |
| `end` | where to end in the file, passed to GGIRread::readAxivity |
| `tz` | time zone for the data `time`, passed to `desiredtz` argument in GGIRread::readAxivity. If NULL or `""`, no time conversion is done. |
| `...` | additional arguments to pass to `GGIRread::readAxivity()` |
| `apply_tz` | turn the `time` column into a `POSIXct` and apply the timezone |
| `verbose` | print diagnostic messages, higher values = more verbosity. |

## Value

A `tibble` with attributes of a header, sample rate, and transformations

## Examples

```
data = acti_read_cwa(acti_example_cwa())
data = acti_read_cwa(
  acti_example_cwa(),
  tz = NULL,
  apply_tz = FALSE,
  verbose = FALSE
)
```

---

`acti_read_genea` *Read Genea Binary File*

---

## Description

Reads binary files from the original 'Genea' accelerometer (Unilever Discover, UK) using `GGIRread::readGenea()`. These are not 'GENEActiv' files, see `acti_read_geneactiv()` for those.

## Usage

```
acti_read_genea(
  path,
  start = 0,
  end = Inf,
  tz = "UTC",
  ...,
  apply_tz = TRUE,
  verbose = TRUE
)

acti_read_genea_header(path)
```

## Arguments

| | |
|---|---|
| `path` | Path to Genea binary file, optionally compressed. |
| `start` | where to start in the file (page number or timestamp), passed to GGIRread::readGenea |
| `end` | where to end in the file (page number or timestamp), passed to GGIRread::readGenea. `Inf` reads to the end of the file. |
| `tz` | time zone of the device clock. The clock time recorded by the device is interpreted as being in `tz`. If `NULL` or `""`, `UTC` is used. |
| `...` | not used |
| `apply_tz` | turn the `time` column into a `POSIXct` in time zone `tz`. If `FALSE`, `time` is left as seconds since 1970-01-01 (device clock time). |
| `verbose` | print diagnostic messages |

## Value

A `tibble` with columns `time`, `X`, `Y`, `Z` (in g) and attributes of a header, sample rate, and transformations

## Examples

```
data = acti_read_genea(acti_example_genea())
header = acti_read_genea_header(acti_example_genea())
```

---

`acti_read_geneactiv` *Read GENEActiv Binary File*

---

## Description

Read GENEActiv Binary File

## Usage

```
acti_read_geneactiv(
  path,
  start = 0,
  end = Inf,
  tz = "UTC",
  ...,
  apply_tz = TRUE,
  verbose = TRUE
)

acti_read_geneactiv_header(path)
```

## Arguments

| | |
|---|---|
| `path` | Path to GENEActiv binary file |
| `start` | where to start in the file, passed to GGIRread::readGENEActiv |
| `end` | where to end in the file, passed to GGIRread::readGENEActiv |
| `tz` | time zone for the data `time`, passed to `desiredtz` argument in GGIRread::readGENEActiv. If NULL or `""`, no time conversion is done. |
| `...` | additional arguments to pass to `GGIRread::readGENEActiv()` |
| `apply_tz` | turn the `time` column into a `POSIXct` in time zone `tz`. If `FALSE`, `time` is left as seconds since 1970-01-01. |
| `verbose` | print diagnostic messages, higher values = more verbosity. |

## Value

A `tibble` with attributes of a header, sample rate, and transformations

## Examples

```
data = acti_read_geneactiv(acti_example_geneactiv())
data = acti_read_geneactiv(
  acti_example_geneactiv(),
  tz = NULL,
  apply_tz = FALSE,
  verbose = FALSE
)
header = acti_read_geneactiv_header(acti_example_geneactiv())
```

---

`acti_read_gt3x` *Read GT3X file*

---

## Description

Read GT3X file

## Usage

```
acti_read_gt3x(
  path,
  asDataFrame = TRUE,
  imputeZeroes = TRUE,
  verbose = TRUE,
  ...,
  digits = 3L,
  fill_zeroes = TRUE,
  apply_tz = FALSE,
  check_attributes = TRUE,
  tz = "GMT"
)

acti_info_gt3x(path, ...)
```

## Arguments

| | |
|---|---|
| `path` | Path to gt3x file |
| `asDataFrame` | convert to an `activity_df`, see `as.data.frame.activity` |
| `imputeZeroes` | Impute zeros in case there are missingness? Default is `TRUE`. If `FALSE`, the time series will be incomplete in case there is missingness. |
| `verbose` | print diagnostic messages, higher values = more verbosity. |
| `...` | additional arguments to pass to `read.gt3x::read.gt3x()` |
| `digits` | number of digits to round the acceleration values to, passed to `read.gt3x::read.gt3x()` (requires `read.gt3x >= 1.3.0`). The default `3L` matches both `read.gt3x::read.gt3x()` and ActiLife, so the data agree with ActiLife exports. Set to `NULL` to use the `read.gt3x::read.gt3x()` default. If you will resample the data (e.g. to 30 Hz |

for activity counts), read with `digits` of at least `6L` (e.g. `8L`), resample, and then round to 3 digits: rounding to 3 before resampling loses precision, and rounding values already rounded to 5 digits rounds twice (e.g. `0.1234996` to `0.12350` to `0.124`).

`fill_zeroes` Rows with all zeros will be filled in with the last observation carried forward as is done with ActiLife. Recommended

`apply_tz` Apply the timezone from the header `TimeZone` attribute, if available

`check_attributes`
Check that the attributes are included This is a sanity check, including checking that `sample_rate` is in the attributes.

`tz` timezone to display the data in when `apply_tz = TRUE`. The data read in via `read.gt3x::read.gt3x()` says the timezone is GMT, but the time values are the device's local clock time. When `apply_tz = TRUE`, the header `TimeZone` offset is used to recover the true time, which is then shown in `tz` (the default `"GMT"` gives UTC times). Set to `NULL` to keep the fixed-offset zone from the header (e.g. `"Etc/GMT+4"` for `-04:00`), or UTC if the offset is not a whole number of hours.

## Value

A `data.frame`

## Examples

```
library(actiread)
data = acti_read_gt3x(acti_example_gt3x())
data = acti_read_gt3x(
  acti_example_gt3x(),
  tz = NULL,
  apply_tz = FALSE,
  verbose = FALSE,
  fill_zeroes = FALSE
)
info = acti_info_gt3x(acti_example_gt3x())
```

---

`acti_read_parmay_matrix`
*Read Parmay Tech Matrix Binary File*

---

## Description

Reads `.bin`/`.BIN` files from Parmay Tech Matrix devices using `GGIRread::readParmayMatrix()`.

## Usage

```
acti_read_parmay_matrix(
  path,
  start = 1,
  end = NULL,
  tz = "UTC",
  ...,
  apply_tz = TRUE,
  verbose = TRUE
)

acti_read_parmay_matrix_header(path)
```

## Arguments

| | |
|---|---|
| path | Path to Matrix binary file, optionally compressed. |
| start | index of the first packet to read, passed to GGIRread::readParmayMatrix |
| end | index of the last packet to read, passed to GGIRread::readParmayMatrix. NULL reads all packets. |
| tz | time zone for the data time, passed to desiredtz in GGIRread::readParmayMatrix. If NULL or "", no time conversion is done. |
| ... | additional arguments to pass to GGIRread::readParmayMatrix(), such as read_gyro, read_temp, or read_heart |
| apply_tz | turn the time column into a POSIXct in time zone tz. If FALSE, time is left as seconds since 1970-01-01. |
| verbose | print diagnostic messages |

## Value

A tibble with columns time, X, Y, Z (in g), any other sensors read, and attributes of a header, sample rate, quality-control log (qc_log), and transformations

## Examples

```
data = acti_read_parmay_matrix(acti_example_parmay_matrix())
header = acti_read_parmay_matrix_header(acti_example_parmay_matrix())
```

---

# acti_read_sensorlog *Read SensorLog Data*

---

## Description

Read SensorLog Data

## Usage

```
acti_read_sensorlog(path, verbose = TRUE, robust = FALSE)

acti_sensorlog_csv_spec()

acti_sensorlog_csv_colnames_mapping()
```

## Arguments

| | |
|---|---|
| path | A character vector of SensorLog files, usually from unzipping the file, or a zip file of SensorLog files |
| verbose | print diagnostic messages. Either logical or integer, where higher values are higher levels of verbosity. |
| robust | if TRUE then acti_rewrite_sensorlog_csv is run on the data to try to fix any shifts with the data. |

## Value

A data.frame of data

## Examples

```
library(actiread)
path = acti_example_sensorlog_file()
df = acti_read_sensorlog(path)
head(df)
```

---

# acti_require_actipy *Require* actipy *for* reticulate

---

## Description

Require actipy for reticulate

## Usage

```
acti_require_actipy(..., python_version = "3.9")
```

## Arguments

| | |
|---|---|
| ... | additional arguments to pass to reticulate::py_require |
| python_version | version of python passed to reticulate::py_require |

## Value

invisible NULL

## acti_rewrite_sensorlog_csv *Rewrite a CSV that may have issues with export from SensorLog*

### Description

Rewrite a CSV that may have issues with export from SensorLog

### Usage

```
acti_rewrite_sensorlog_csv(
  path,
  outfile = tempfile(fileext = ".csv"),
  verbose = TRUE
)
```

### Arguments

| | |
|---|---|
| path | Input CSV file |
| outfile | Output CSV file |
| verbose | Print Diagnostic messages |

### Value

A file path to the new CSV

### Examples

```
sl_file = actiread::acti_example_sensorlog_file()
tfile = tempfile()
files = utils::unzip(sl_file, exdir = tfile)
result = actiread::acti_rewrite_sensorlog_csv(files)
```

## acti_sensorlogger_location_colnames_mapping *Read SensorLogger Data*

### Description

Read SensorLogger Data

## Usage

```
acti_sensorlogger_location_colnames_mapping()

acti_sensorlogger_location_spec()

acti_read_sensorlogger_location(path, ...)

acti_read_sensorlogger(path, verbose = TRUE, ...)

acti_read_sensorlogger_general(path, ..., verbose = TRUE)
```

## Arguments

| | |
|---|---|
| path | A character vector of SensorLogger files, usually from unzipping the file, or a zip file of SensorLogger files |
| ... | additional arguments to pass to readr::read_csv(). If verbose = FALSE, then progress = FALSE and show_col_types = FALSE, unless otherwise overridden |
| verbose | print diagnostic messages. Either logical or integer, where higher values are higher levels of verbosity. |

## Value

A data.frame of data if a single CSV file is given, otherwise a named list of data.frames, one per sensor file (always a list for zip files)

---

# tzoffset_to_tz *Transform timezone offset to timezone*

---

## Description

Converts UTC offsets such as "-05:00" (5 hours behind UTC) into the matching fixed-offset Olson time zone. Note that Etc/GMT zones use an inverted sign convention, so "-05:00" is "Etc/GMT+5".

## Usage

```
tzoffset_to_tz(x)
```

## Arguments

| | |
|---|---|
| x | A character vector of offsets in the form "+HH:MM" or "+HH:MM:SS" |

## Value

A character vector

## Examples

```
tzoffset_to_tz(c("+00:00", "-05:00", "+01:00"))
```

# Index


acti_convert_sensorlogger_time, 2
acti_decompress_file, 3
acti_example_cwa *(*acti_example_gt3x*)*, 3
acti_example_genea *(*acti_example_gt3x*)*, 3
acti_example_geneactiv *(*acti_example_gt3x*)*, 3
acti_example_gt3x, 3
acti_example_parmay_matrix *(*acti_example_gt3x*)*, 3
acti_example_sensorlog_file *(*acti_example_gt3x*)*, 3
acti_example_sensorlogger_file *(*acti_example_gt3x*)*, 3
acti_example_sensorlogger_location_file *(*acti_example_gt3x*)*, 3
acti_info_gt3x *(*acti_read_gt3x*)*, 9
acti_py_read_cwa, 4
acti_read, 5
acti_read_cwa, 6
acti_read_cwa_header *(*acti_read_cwa*)*, 6
acti_read_genea, 7
acti_read_genea_header *(*acti_read_genea*)*, 7
acti_read_geneactiv, 8
acti_read_geneactiv(), *7*
acti_read_geneactiv_header *(*acti_read_geneactiv*)*, 8
acti_read_gt3x, 9
acti_read_parmay_matrix, 10
acti_read_parmay_matrix_header *(*acti_read_parmay_matrix*)*, 10
acti_read_sensorlog, 11
acti_read_sensorlogger *(*acti_sensorlogger_location_colnames_mapping*)*, 13
acti_read_sensorlogger_general *(*acti_sensorlogger_location_colnames_mapping*)*, 13
acti_read_sensorlogger_location *(*acti_sensorlogger_location_colnames_mapping*)*, 13
acti_require_actipy, 12
acti_rewrite_sensorlog_csv, *12*, 13
acti_sensorlog_csv_colnames_mapping *(*acti_read_sensorlog*)*, 11
acti_sensorlog_csv_spec *(*acti_read_sensorlog*)*, 11
acti_sensorlogger_location_colnames_mapping, 13
acti_sensorlogger_location_spec *(*acti_sensorlogger_location_colnames_mapping*)*, 13

GGIRread::readAxivity, *5, 6*
GGIRread::readAxivity(), *6*
GGIRread::readGenea, *7*
GGIRread::readGenea(), *7*
GGIRread::readGENEActiv, *8*
GGIRread::readGENEActiv(), *8*
GGIRread::readParmayMatrix, *11*
GGIRread::readParmayMatrix(), *10, 11*

read.gt3x::read.gt3x(), *9, 10*
readr::read_csv(), *14*
reticulate::py_require, *12*

tzoffset_to_tz, 14