Package 'cgmanalysis'

February 3, 2025

Title Clean and Analyze Continuous Glucose Monitor Data

Version 3.0.2

Description This code provides several different functions for cleaning and analyzing continuous glucose monitor data. Currently it works with 'Dexcom', 'iPro 2', 'Diasend', 'Libre', or 'Carelink' data. The cleandata() function takes a directory of CGM data files and prepares them for analysis. cgmvariables() iterates through a directory of cleaned CGM data files and produces a single spreadsheet with data for each file in either rows or columns. The column format of this spreadsheet is compatible with REDCap data upload. cgmreport() also iterates through a directory of cleaned data, and produces PDFs of individual and aggregate AGP plots. Please visit <https://github.com/childhealthbiostatscore/R-Packages/> to download the new-user guide.

Depends R (>= 4.0.0)

License CC0

Encoding UTF-8

RoxygenNote 7.3.2

Imports ggplot2, parsedate, lubridate, pracma, zoo, pastecs, tools, readxl, readr, XML, MESS

NeedsCompilation no

Author Tim Vigers [aut, cre]

Maintainer Tim Vigers <timothy.vigers@cuanschutz.edu>

Repository CRAN

Date/Publication 2025-02-03 19:30:10 UTC

Contents

cgmreport *Generate AGP*

Description

This function takes a directory of cleaned CGM data and generates two aggregate AGPs for all of the files combined. One aggregate AGP is produced using Tukey smoothing, and the other uses ggplot's built in LOESS smoothing function. The function also produces an AGP plot with each unique subject plotted individually on the same graph (using LOESS smoothing).

Usage

```
cgmreport(inputdirectory, outputdirectory = tempdir(), tz = "UTC",
yaxis = c(0,400))
```

Arguments

`inputdirectory`	The directory containing all cleaned, formatted CGM data to be analyzed.
`outputdirectory`	The directory where plot PDF files should be written.
`tz`	The time zone in which the data were recorded.
`yaxis`	The range of the yaxis in mg/dL.

Value

Aggregate and per subject AGP reports based on all of the cleaned CGM data in the input directory.

Examples

```
cgmreport(system.file("extdata", "Cleaned", package = "cgmanalysis"))
```

cgmvariables *Calculate CGM Variables*

Description

This function takes cleaned CGM data and returns clinically relevant measures (e.g. percent time spent over 140, MAGE, MODD, etc.).

Usage

```
cgmvariables(inputdirectory,
outputdirectory = tempdir(),
outputname = "REDCap Upload",
customintervals = list(c(70, 140), c(180, 250), c(250, 400)),
aboveexcursionlength = 35,
belowexcursionlength = 10,
magedef = "1sd",
congan = 1,
daystart = 6,
dayend = 22,
id_filename = F,
format = "rows",
printname = F,
unit = "mg/dL"
)
```

Arguments

inputdirectory	The directory containing cleaned CSV files for analysis.
outputdirectory	The directory where you would like the results spreadsheet to be written.
outputname	The name of the file containing final CGM variables (without the file extension).
customintervals	A list of custom blood glucose intervals. Minutes and percent time below the lower bound, in the specified range, and above the upper bound are calculated for each interval in the list. Number of excursions below the lower bound and above the upper bound are also calculated for each interval.
aboveexcursionlength	The number of minutes blood sugar must be above threshold to count an excursion.
belowexcursionlength	The number of minutes blood sugar must be below threshold to count an excursion.
magedef	How large an excursion needs to be in order to count in the MAGE calculation (e.g. greater than 1 standard deviation).
congan	CONGA interval in hours.
daystart	The numeric hour at which daytime should start (e.g. to start counting day time at 6:00am, set daystart = 6).
dayend	The numeric hour at which daytime should end (this parameter uses military time, so to stop counting day time at 10:00pm, set dayend = 22).
id_filename	If true, the file name will be used for subject ID rather than the ID contained in the data.
format	Whether observations are in rows or columns.
printname	Whether or not to print each file name (for troubleshooting).

`unit` Default unit is mg/dL. Any other value will multiply the "sensorglucose" column by 18 (i.e. the package assumes that unit != "mg/dL" implies that the units are mmol/L). All relevant results are output in mg/dL.

Details

All files must be saved as a csv, and must have three columns, the first of which contains the subject ID in the first cell and date of CGM placement in the second (see example files). The names of the columns must be "subjectid" "timestamp" and "sensorglucose" (without quotes) respectively. Files can be cleaned and formatted using this package's cleandata() function.

Value

A data frame containing calculated CGM variables, with each column representing one CGM file.

Examples

```
cgmvariables(system.file("extdata", "Cleaned", package = "cgmanalysis"))
```

---

cleandata *Clean CGM Data*

Description

This function returns cleaned CGM files for analysis. Files must not be edited, and should be saved in the original format. If any files need to be edited manually, please save them in the format specified by the cgmvariables() function. If this function is unable to read your unedited CGM data, it may help to save your data in the format above.

Usage

```
cleandata(inputdirectory,
outputdirectory = tempdir(),
removegaps = TRUE,
gapfill = TRUE,
maximumgap = 20,
id_filename = F,
verbose = F,
unit = "mg/dL")
```

Arguments

`inputdirectory` The directory containing CSV files for cleaning prior to analysis.

`outputdirectory` The directory where cleaned CSV files will be written.

removegaps	Determines whether the data are cleaned or not. If set to TRUE, any gaps in the data will be removed along with the 24 hours of data containing the gap(s). The tail end of the data will also be trimmed to ensure the timeseries is in discrete 24 hour chunks.
gapfill	If set to TRUE (and if removegaps = TRUE), gaps smaller than or equal to maximumgap will be interpolated rather than removed.
maximumgap	Allows the user to determine the longest data gap (in minutes) that will be interpolated.
id_filename	If true, the file name will be used for subject ID rather than the ID contained in the data.
verbose	If true, each file name will print as the program runs for troubleshooting purposes.
unit	Default unit is mg/dL. Any other value will multiply the "sensorglucose" column by 18 (i.e. the package assumes that unit != "mg/dL" implies that the units are mmol/L).

Details

Because Diasend data is exported in an Excel document containing multiple tabs, the CGM data must be in the first tab in order to be read effectively.

Examples

```
## Not run:
cleandata(system.file("extdata", "De-identified",
  package = "cgmanalysis"
))

## End(Not run)
```

Index