RESEARCH ARTICLE

cgmanalysis: An R package for descriptive analysis of continuous glucose monitor data

Tim Vigers[1,2,3], Christine L. Chan[1], Janet Snell-Bergeon[2], Petter Bjornstad[1], Philip S. Zeitler[1], Gregory Forlenza[2], Laura Pyle[1,2,3]*

1 Section of Pediatric Endocrinology, University of Colorado School of Medicine, Aurora, Colorado, United States of America, **2** Barbara Davis Center, University of Colorado School of Medicine, Aurora, Colorado, United States of America, **3** Department of Biostatistics and Informatics, Colorado School of Public Health, Aurora, Colorado, United States of America

* Laura.pyle@ucdenver.edu

Citation: Vigers T, Chan CL, Snell-Bergeon J, Bjornstad P, Zeitler PS, Forlenza G, et al. (2019) cgmanalysis: An R package for descriptive analysis of continuous glucose monitor data. PLoS ONE 14 (10): e0216851. https://doi.org/10.1371/journal.pone.0216851

Editor: Kathleen E. Bethin, University at Buffalo, UNITED STATES

Received: April 26, 2019

Accepted: September 22, 2019

Published: October 11, 2019

Data Availability Statement: All data can be found in the Figshare repository at the following link: https://figshare.com/projects/R_Functions_for_Analysis_of_Continuous_Glucose_Monitor_Data/64973

Funding: Funding sources: NIH (www.nih.gov) grant DK094712-04 (GF) and NIH grant 5K12DK094712-04 and Cystic Fibrosis Foundation (www.cff.org) Therapeutics grants CHAN16A0 and CHAN16GE0 (CC). The funders had no role in study

Abstract

Continuous glucose monitoring (CGM) is an essential part of diabetes care. Real-time CGM data are beneficial to patients for daily glucose management, and aggregate summary statistics of CGM measures are valuable to direct insulin dosing and as a tool for researchers in clinical trials. Yet, the various commercial systems still report CGM data in disparate, non-standard ways. Accordingly, there is a need for a standardized, free, open-source approach to CGM data management and analysis. A package titled cgmanalysis was developed in the free programming language R to provide a rapid, easy, and consistent methodology for CGM data management, summary measure calculation, and descriptive analysis. Variables calculated by our package compare well to those generated by various CGM software, and our functions provide a more comprehensive list of summary measures available to clinicians and researchers. Consistent handling of CGM data using our R package may facilitate collaboration between research groups and contribute to a better understanding of free-living glucose patterns.

Introduction

Continuous glucose monitoring (CGM) technology has transformed diabetes care over the past 15 years by allowing clinicians to measure free-living glucose patterns. During this period, CGM use has increased from $<$ 5% of patients to almost 50% in some age groups [1]. With recent reports detailing the benefits of CGM time in range metrics as predictive of long-term vascular outcomes [2] and as an indicator of glucose management or estimated hemoglobin A1c (HbA1c) [3], CGM use will likely continue to increase in both research and clinical settings. Despite the increasing use of CGM for treatment and research, a standardized, free, open-source approach to data management and analysis is lacking [4].

CGM manufacturers use proprietary algorithms to create reports and calculate summary measures for patients and clinicians. As a result, it may be difficult to compare results obtained using different CGM devices and to understand the sources of variability

design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing interests: The authors have declared that no competing interests exist.

that could influence CGM outcomes. In addition, research questions may require summary measures that are not available in accompanying reports (e.g., use of a different cut-point for hyperglycemia). Furthermore, use of the summary values provided by each CGM platform sometimes requires that data be entered by hand into a database or spreadsheet prior to analysis. This is a time-consuming and error prone process that will benefit from automation. The use of a free and open source program to summarize raw sensor glucose values will enable researchers to define their own variables of interest and standardize calculation of summary measures across different CGM devices.

There have already been a few attempts to develop such systems, including the EasyGV macro-enabled Excel workbook [5], AGP Report (agpreport.org), and Tidepool (tidepool.org). However, there are reports suggesting that EasyGV poorly matches other calculations of mean amplitude of glycemic excursion (MAGE) [6], and it does not permit the various definitions of a significant excursion (i.e. greater than 1 standard deviation (SD), 2 SDs, etc.). Although Tidepool appears to be an excellent option for patients and clinicians, it is not free for use in research, and many smaller investigator-initiated studies cannot afford the additional expense. Also, their open source code requires significant coding knowledge in multiple programming languages which limits accessibility and widespread

Table 1. Summary measures of glycemia.

CGM Variable	Definition
percent_cgm_wear	The number of sensor readings as a percentage of the number of potential readings (given time worn).
average_sensor	Mean of all sensor glucose values
estimated_a1c	Estimated HbA1c based on the equation: (46.7 + average glucose in mg/dL) / 28.7 [1]
gmi	Glucose management indicator based on the equation: 3.31 + (0.02392 × average glucose in mg/dL)[7]
q1_sensor	First quartile sensor glucose value
median_sensor	Median sensor glucose value
q3_sensor	Third quartile sensor glucose value
standard_deviation	Standard deviation of all sensor glucose values
cv	Coefficient of variation of all sensor glucose values (SD/mean)
min_sensor	Minimum of all sensor glucose values
max_sensor	Maximum of all sensor glucose values
excursions_over_***	The number of local glucose peaks with an amplitude greater than *** mg/dL
min_spent_over_***	The total length of time that sensor glucose was at or above *** mg/dL
percent_time_over_***	Minutes spent above *** mg/dL, as a percentage of the total time CGM was worn
avg_excur_over_***_per_day	The number of glucose peaks above *** mg/dL averaged per 24-hour period of CGM wear
min_spent_under_**	The total length of time that sensor glucose was at or below ** mg/dL
percent_time_under_**	Minutes spent below ** mg/dL, as a percentage of the total time CGM was worn
min_spent_70_180	Minutes spent in the range 70–180 mg/dL (inclusive)
percent_time_70_180	Minutes spent in the range 70–180 mg/dL (inclusive), as a percentage of the total time CGM was worn
daytime_***	*** of all sensor glucose values during specified daytime hours
nighttime_***	*** of all sensor glucose values during specified nighttime hours
auc	Approximate area under the sensor glucose curve, calculated using the trapezoidal rule
r_mage	MAGE calculated according to Baghurst's algorithm
j_index	Calculated based on the equation: 0.324 × (average glucose in mg/dL + standard deviation of glucose levels)^2[11]
conga	Continuous overall net glycemic action, default n = 1 hour[11]
modd	Mean of daily differences
lbgi	Low blood glucose index
hbgi	High blood glucose index

https://doi.org/10.1371/journal.pone.0216851.t001 use. Finally, Zhang et al. [7] released the CGManalyzer package for R; however, the package was removed from the CRAN repository because problems with the software were not corrected.

To address this need, we have developed a package written entirely in the statistical programming language R (R Foundation for Statistical Computing, Vienna, Austria). R software is free and can be obtained at: https://www.r-project.org/. The package currently works with data from Diasend (www.diasend.com), Dexcom (www.dexcom.com), iPro 2 (http://professional.medtronicdiabetes.com/ipro2-professional-cgm), Libre (www.freestylelibre.us), and Carelink (www.medtronicdiabetes.com/products/carelink-personal-diabetes-software), with plans to add support for other platforms as CGM technology advances. Additionally, data can be manually formatted to work with these functions if necessary. The package is available on The Comprehensive R Archive Network (CRAN) under the name 'cgmanalysis' (https://cran.r-project.org/web/packages/cgmanalysis/index.html) and the source code can be found at https://github.com/childhealthbiostatscore/R-Packages, which allows for version control and forking if users need to modify the code to alter functionality. A short user guide (https://github.com/childhealthbiostatscore/

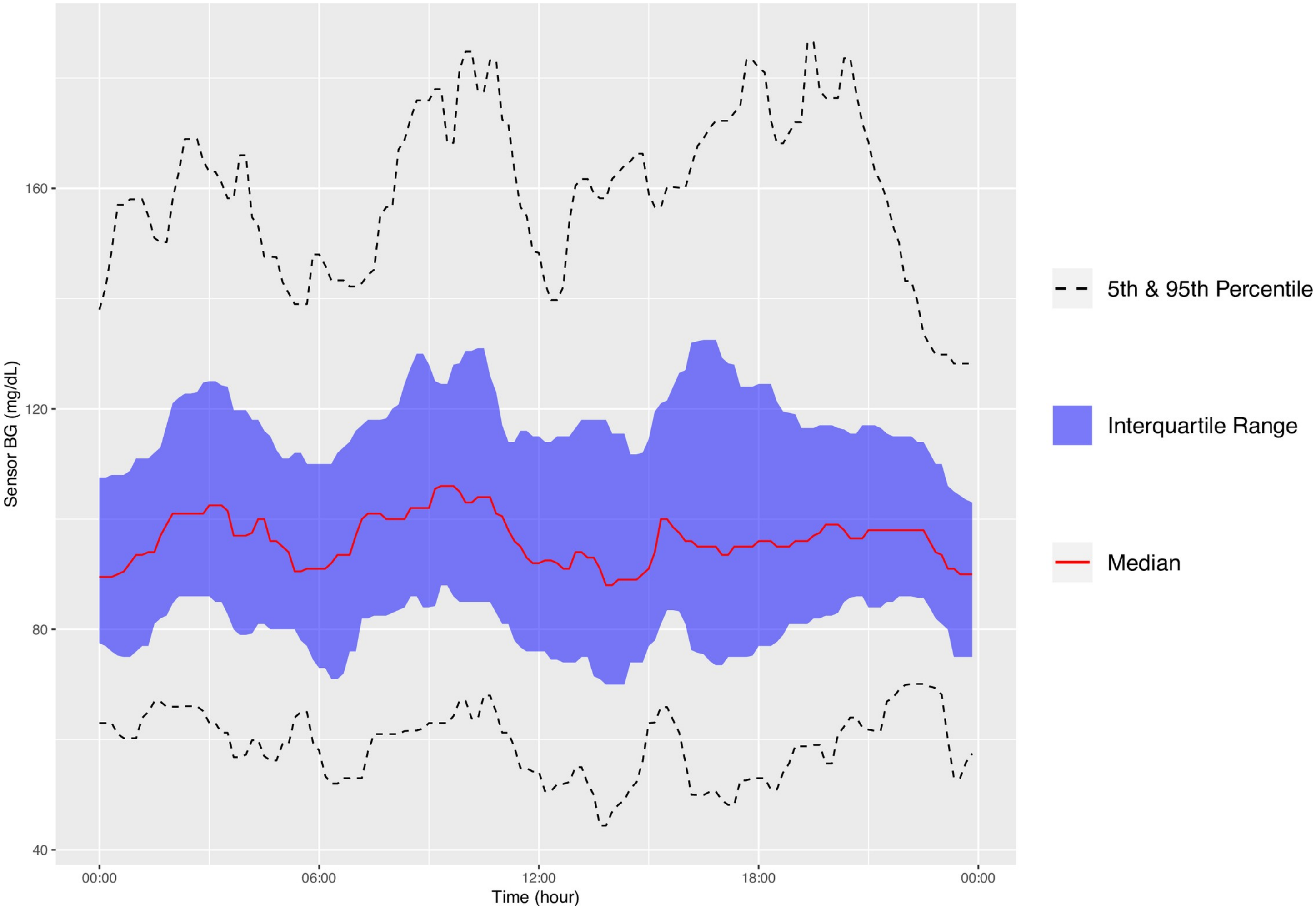

Fig 1. Aggregate Daily Overlay (Tukey Smoothing).

https://doi.org/10.1371/journal.pone.0216851.g001

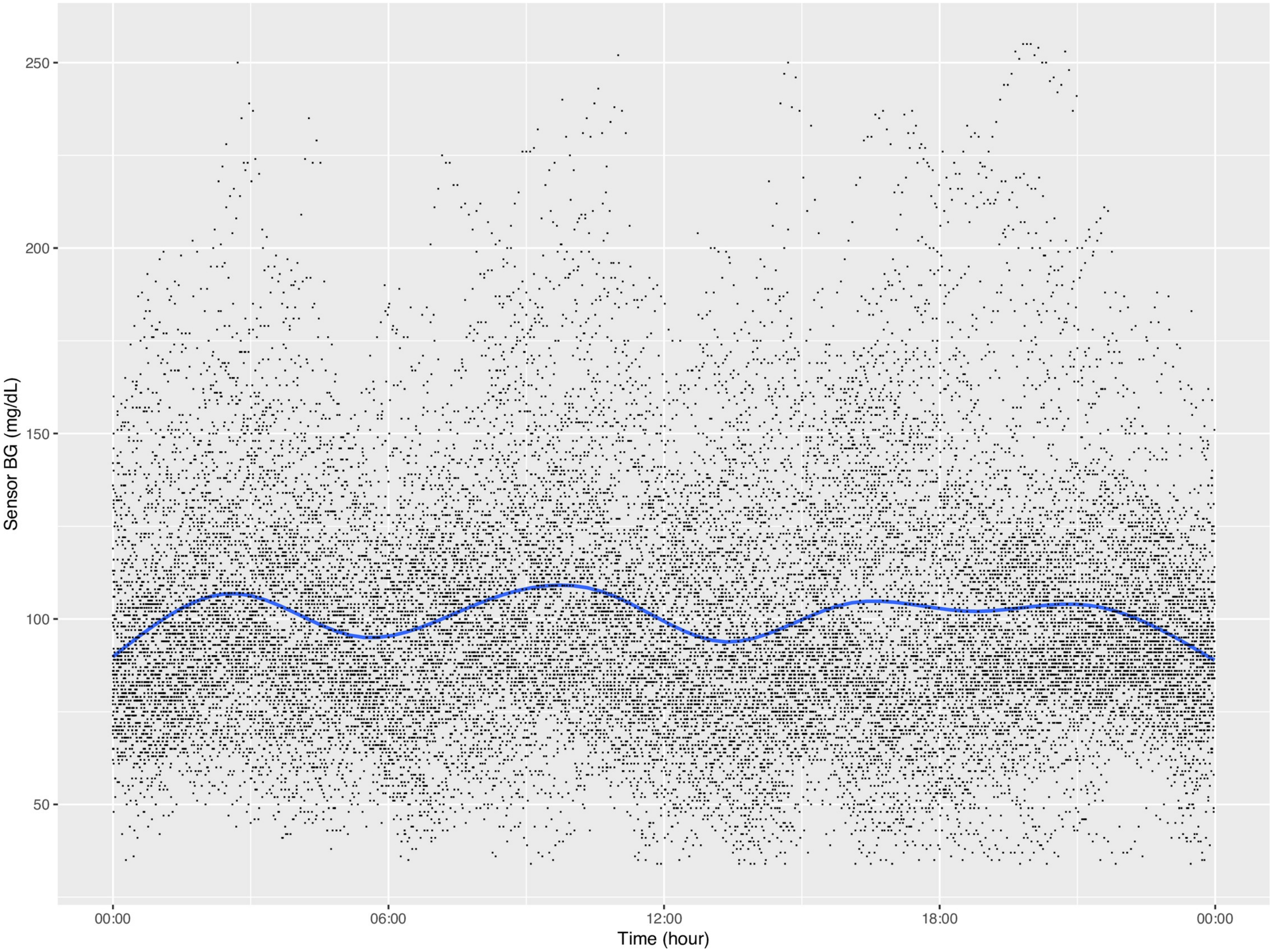

Fig 2. Aggregate Daily Overlay (Loess Smoothing).

https://doi.org/10.1371/journal.pone.0216851.g002

R-Packages/blob/master/CGM%20Analysis/cgmanalysis%20New-User%20Guide.docx) explains how to install and run the software.

Summary measures of glycemia

Although CGM is not a new technology, there is still debate regarding the advantages and disadvantages of various CGM metrics for use in clinical care and as research outcomes. The American Diabetes Association (ADA) recently proposed a set of key metrics for reporting CGM data [8], all of which are calculated by our code, in addition to the glucose management indicator (GMI) [3], time in range [2], and other variables proposed by Hernandez et al. [4]. An easy method to calculate these important summary variables from a variety of sources of CGM data has the potential to contribute to the standardization of the use of these metrics. A list of summary variables produced by our default code is available in **Table 1**. The code can be easily modified to include further variables of interest, to

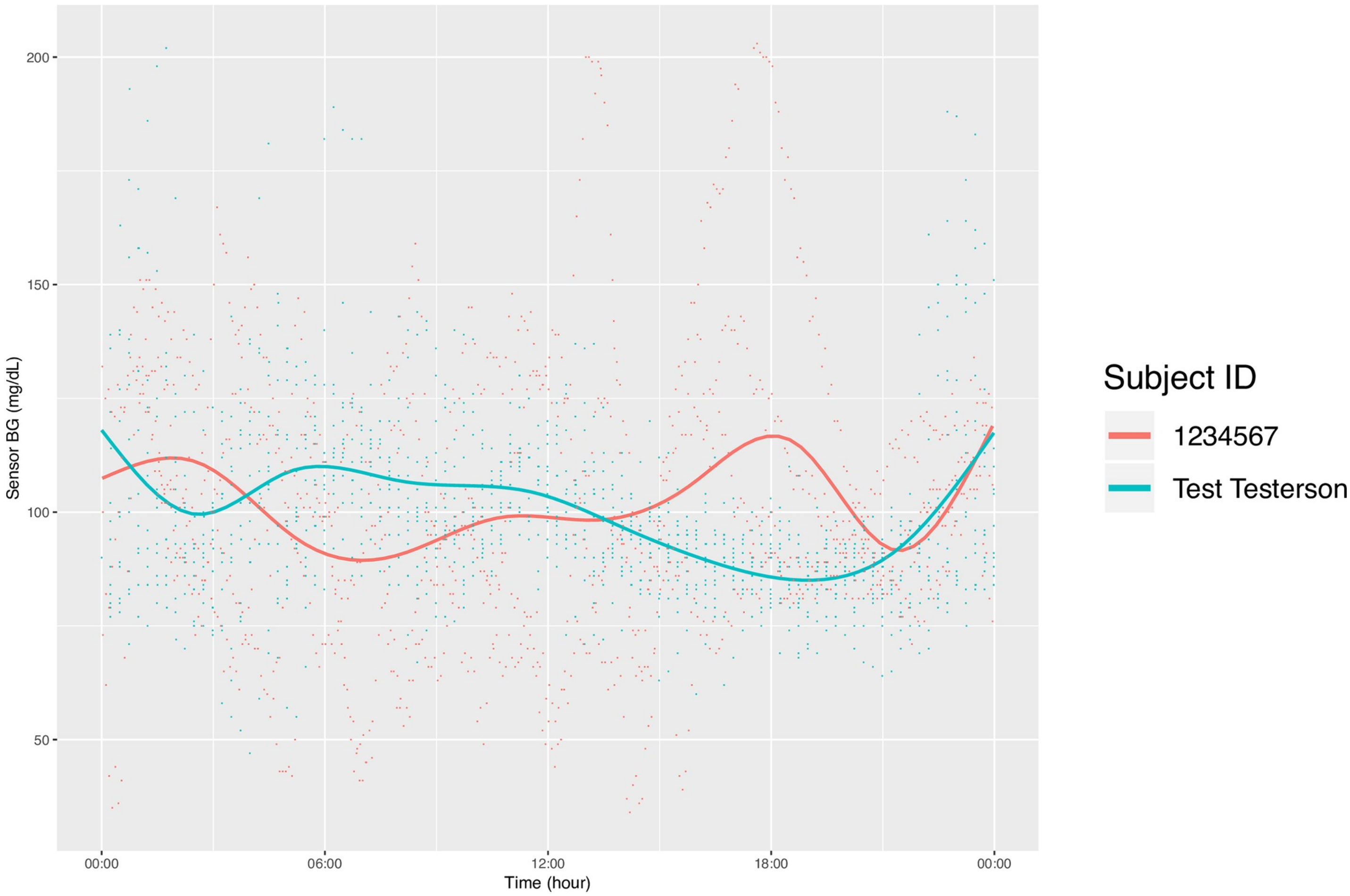

Fig 3. Daily Overlay per Subject (LOESS Smoothing).

https://doi.org/10.1371/journal.pone.0216851.g003

be released in future version updates. Further, because the package is open source, individual users can create their own modifications.

Methods

Package design

Our package consists of three simple functions: cleandata(), cgmvariables(), and cgmreport(). The data cleaning function iterates through a directory of CGM data exports and produces new files that then serve as input to the CGM variable calculator and the CGM report generator. The initial directory can contain files from different sources, as the function identifies the relevant timestamp and glucose values for each file format. By default, the cleaning function will fill in gaps in glucose data less than 20 minutes long using linear interpolation. It will also remove 24-hour periods containing gaps larger than 20 minutes, so that there will be an equal number of daytime and nighttime values, important for calculating some variables, such as AUC. The user can specify a different maximum gap to fill by interpolation and can also choose whether to remove days with larger gaps. For example,

```
    cleandata(“path/to/inputdirectory”,
    “path/to/outputdirectory”)
```

will clean the data using the default settings, while

```
  cleandata(“path/to/inputdirectory”,
  “path/to/outputdirectory”,
  removegaps = FALSE, gapfill = TRUE, maximumgap = 30)
```

will fill in gaps shorter than 30 minutes but will not remove the 24-hour chunks containing larger gaps. Ideally, the CGM data should be exported and then cleaned using this package, and not manually edited. However, if a file does require manual data editing, these functions will work on the three-column format detailed in the package documentation. Examples of data pre- and post-cleaning are available on figshare (https://figshare.com/projects/cgmanalysis_An_R_package_for_descriptive_analysis_of_continuous_glucose_monitor_data/64973) and in the package's "extdata" directory.

Once the data have been cleaned, the CGM variables described in **Table 1** are calculated using the cgmvariables() function. By default, blood glucose must be above a threshold for at least 35 minutes or below a threshold for at least 10 minutes to count as an excursion, but these parameters can be changed by the user if necessary. Likewise, daytime (e.g. for daytime vs. nighttime AUC or maximum glucose) is defined as 6:00 to 22:00 by default, but these can be set depending on user needs. MAGE is calculated using Baghurst's algorithm [9], which we have coded in R. By

Table 2. Summary variable comparisons.

iPro 2 (high excursion defined as > 140 mg/dL for 15 minutes, low defined as < 60 mg/dL for 15 minutes)		
	cgmanalysis	**iPro 2**
# Sensor Values	2000	2000
Highest	282	282
Lowest	70	70
Average	126.87	127
Standard Dev	30.79	31
# High Excursions	31	32
# Low Excursions	0	0
% Time Above 140	24.85	24
% Time Below 60	0	0
Carelink 670G		
	cgmanalysis	Carelink 670G
Average	123.65	124
Standard Dev	37.53	38
Dexcom Clarity		
	cgmanalysis	Dexcom Clarity
Average	175.68	176
Standard Dev	67.10	68
Time in Range	55.66	56
Diasend		
	cgmanalysis	Diasend
# Sensor Values	184	184
Highest	411	411
Lowest	54	54
Average	193.23	193
Standard Dev	89.67	89
% values above 200	44.57	44.57

https://doi.org/10.1371/journal.pone.0216851.t002 default, the function includes blood glucose excursions greater than 1 SD from the mean in calculation of MAGE, but there are options for 1.5 SD and 2 SD as well. For example,

```
cgmvariables(“path/to/inputdirectory”,
“path/to/outputdirectory”)
```

will produce summary measures using the default settings above, while

```
cgmvariables(“path/to/inputdirectory”,
“path/to/outputdirectory”,
daystart = 8, dayend = 23, magedef = “2sd”)
```

will produce summary measures using 2 SD as the threshold for MAGE excursions, and daytime defined as 8:00 to 23:00.

Our code was originally written to produce data tables for upload to a Research Electronic Data Capture (REDCap) database [10], which influenced the selection of variable names in the final output. These names can be changed in the code itself or by simply editing the function's output. These variables are stored in separate columns of a new data frame (the function's output), with each record identified by the patient ID.

In addition to producing calculated variables, our package can also plot CGM data in a few ways. First, the function concatenates all the CGM data in the specified directory into one data table and plots the aggregate data in the style of the standard AGP report (http://www.agpreport.org), the aggregate daily overlay (ADO). This method uses Tukey running median

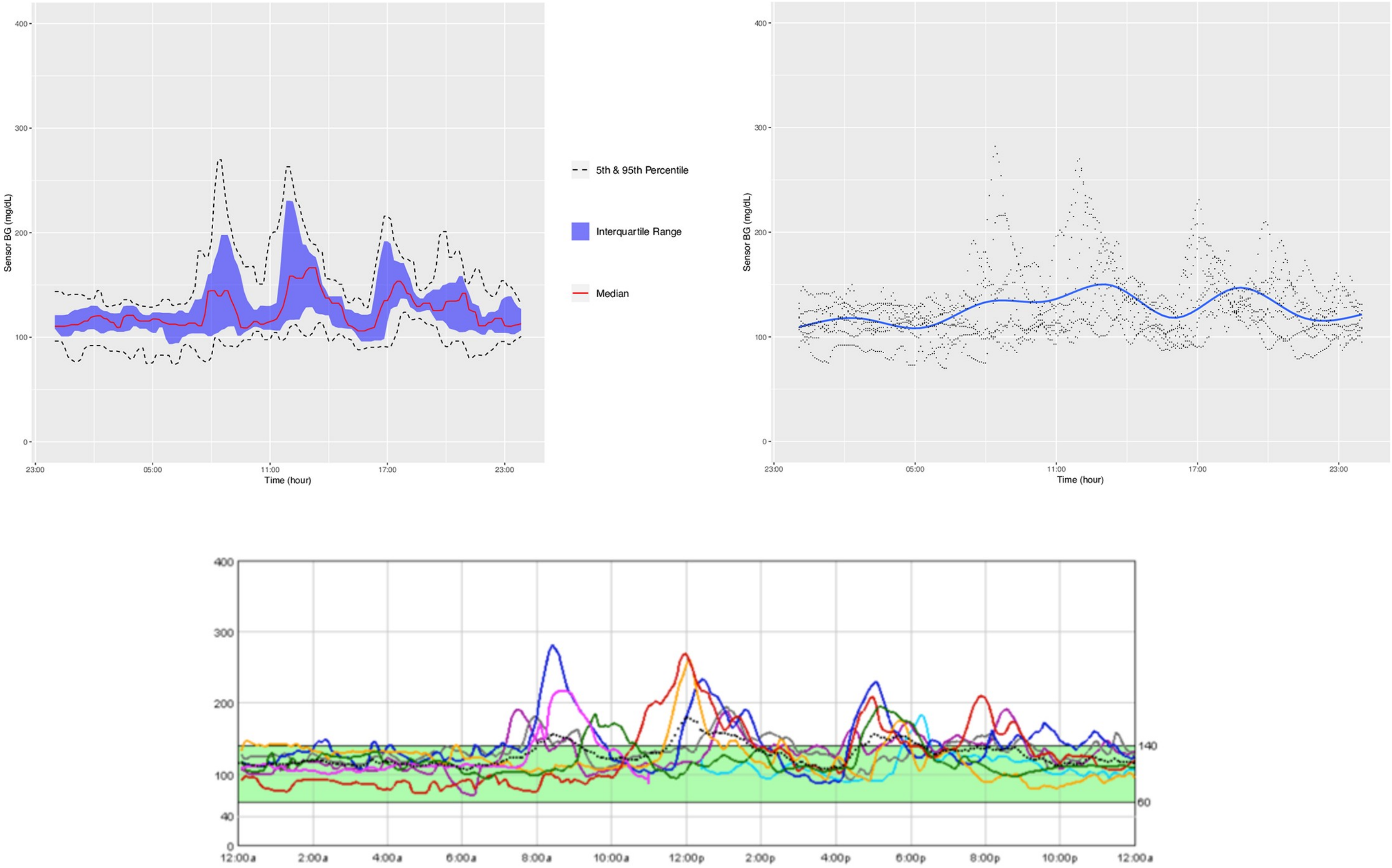

Fig 4. “cgmanalysis” Package Plots Compared to iPro 2 Daily Overlay Clockwise from top left: Aggregate Daily Overlay (Tukey Smoothing), Aggregate Daily Overlay (Loess Smoothing), iPro 2 Daily Overlay.

https://doi.org/10.1371/journal.pone.0216851.g004

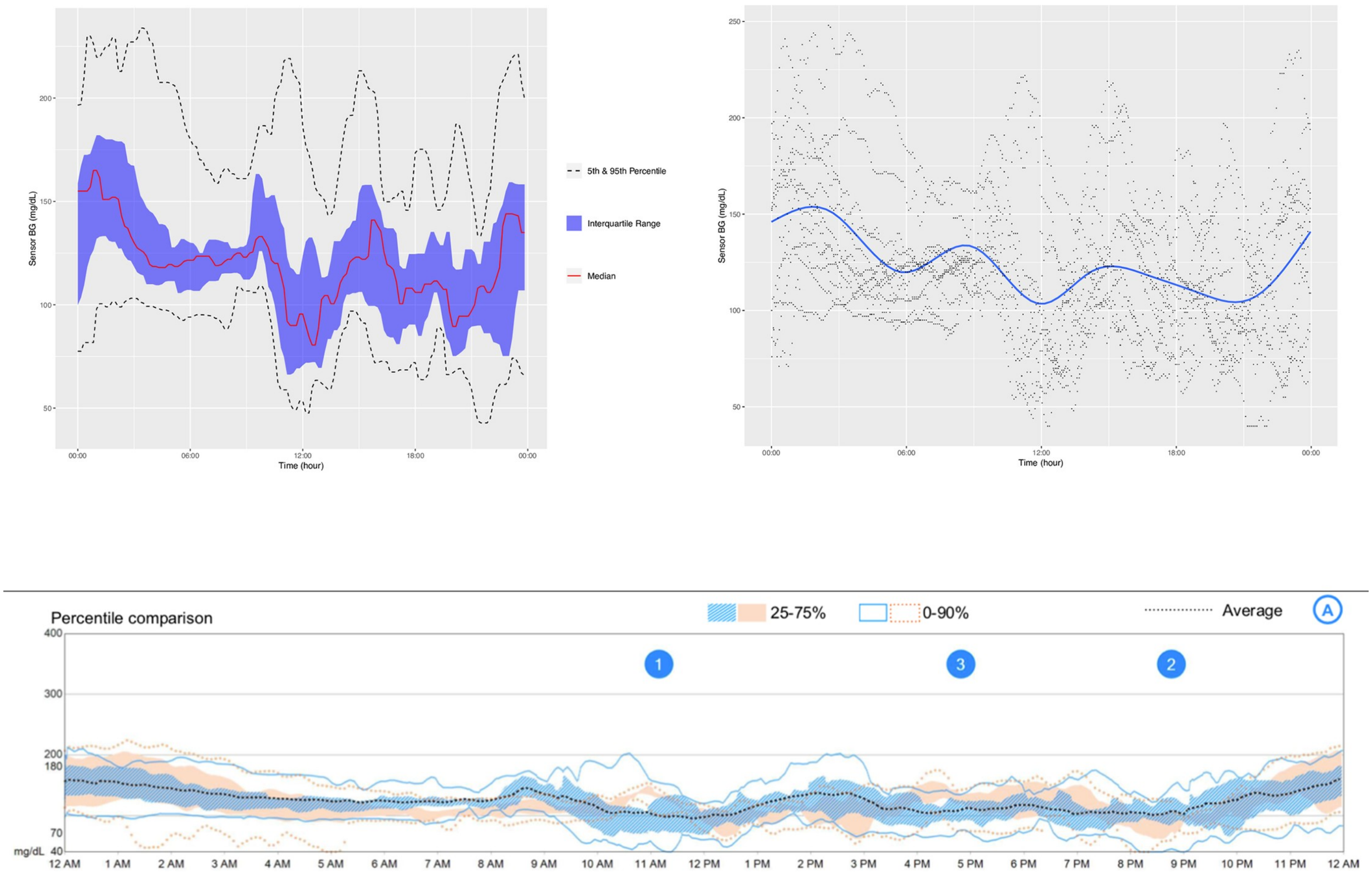

Fig 5. "cgmanalysis" Package Plots Compared to Carelink 670G Daily Overlay Clockwise from top left: Aggregate Daily Overlay (Tukey Smoothing), Aggregate Daily Overlay (Loess Smoothing), Carelink 670G Daily Overlay.

https://doi.org/10.1371/journal.pone.0216851.g005

smoothing [11] after rounding each timepoint to the nearest 10-minute mark, then plots the median, inter-quartile range, and 5 and 95 percentiles at each time of day (with plans to add more options in the future). The package also produces a similar aggregate plot with a Loess-smoothed (locally estimated scatterplot smoothing) average [12–14] overlaid on points representing every single glucose value. For smaller data sets, this type of plot gives a meaningful overview of daily glucose trends. Finally, the third type of plot uses a Loess-smoothed average for each patient with glucose values color-coded by participant. The current default y axis range for each plot is 0–400 mg/dL, but this can be altered manually. For example,

```
cgmreport("path/to/inputdirectory",
"path/to/outputdirectory", yaxis = c(70,300))
```

will produce plots with a y axis range of 70–300 mg/dL.

Comparison of cgmanalysis package and proprietary software

Our functions were compared to proprietary CGM software using clinically collected data from iPro 2, Carelink 670G, Dexcom Clarity, and Diasend. The data were exported from each platform, formatted using the cleandata() function, then summarized using the cgmvariables() and cgmreport() functions. The data were not cleaned prior to plotting and summary variable calculation, and summary variable parameters were altered from default (e.g. defining an

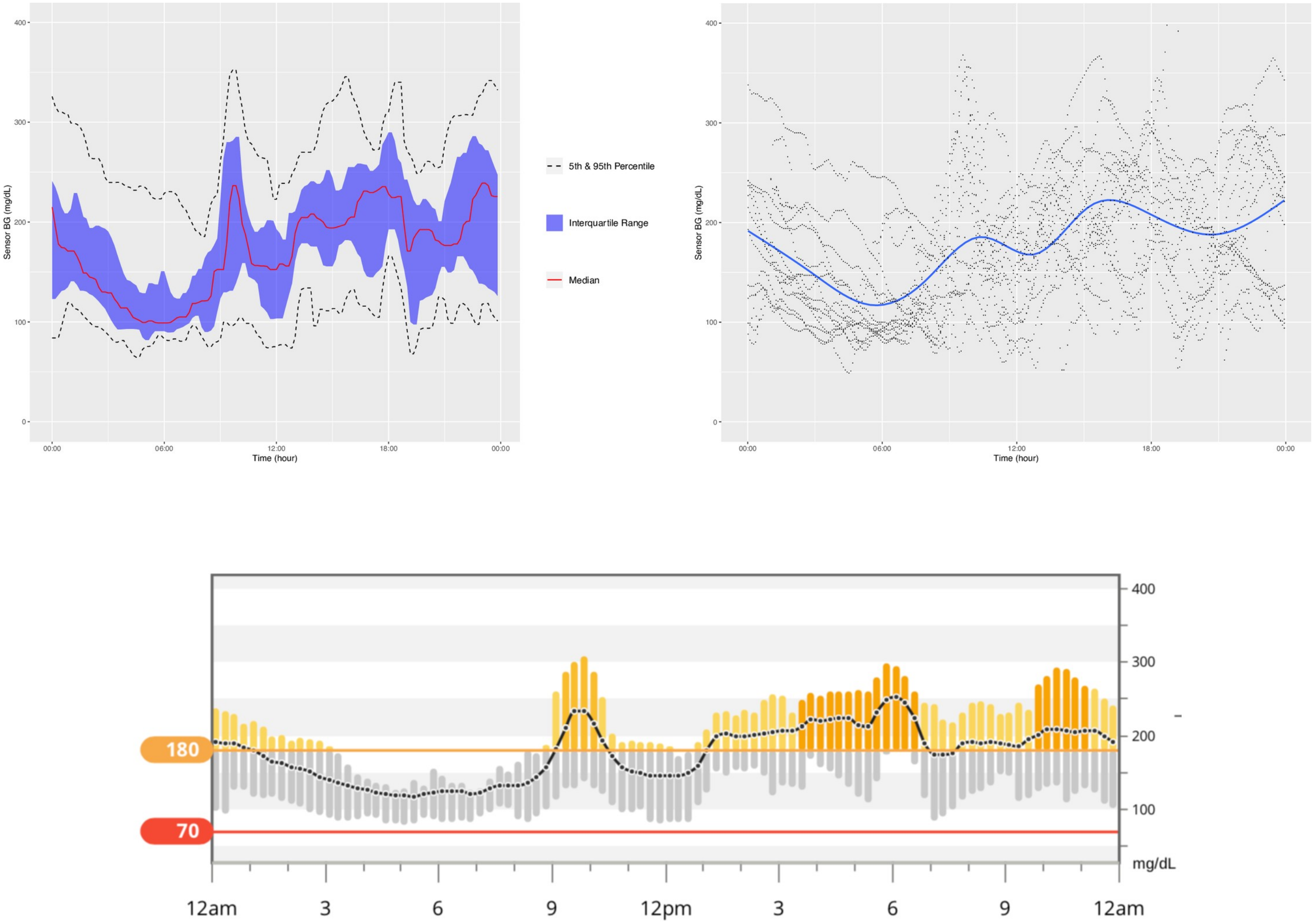

Fig 6. "cgmanalysis" Package Plots Compared to Dexcom Clarity Daily Overlay Clockwise from top left: Aggregate Daily Overlay (Tukey Smoothing), Aggregate Daily Overlay (Loess Smoothing), Dexcom Daily Overlay.

https://doi.org/10.1371/journal.pone.0216851.g006

excursion as 15 minutes above or below threshold for iPro 2 data) in order to better match the CGM results. Because each CGM device provides different and limited summary variables, we were only able to compare a small subset of our package's output and were not able to directly test more complex variables, such as MAGE or CONGA.

Results

Fig 1 is an example of the ADO plot made using approximately 25,000 simulated CGM values, and Fig 2 is the version of the ADO with Loess smoothing, using the same data as in Fig 1. Fig 3 is the patient-specific plot, made with a subset of the simulated data.

Table 2 shows the results of summary variable comparisons between four different proprietary CGM devices and our cgmanalysis package. Most of the differences in these comparisons are small and the result of rounding. Overall the package appears to be capable of reproducing proprietary calculations when run with non-default settings, although in the comparison to the iPro 2, there was a difference of 1 high excursion.

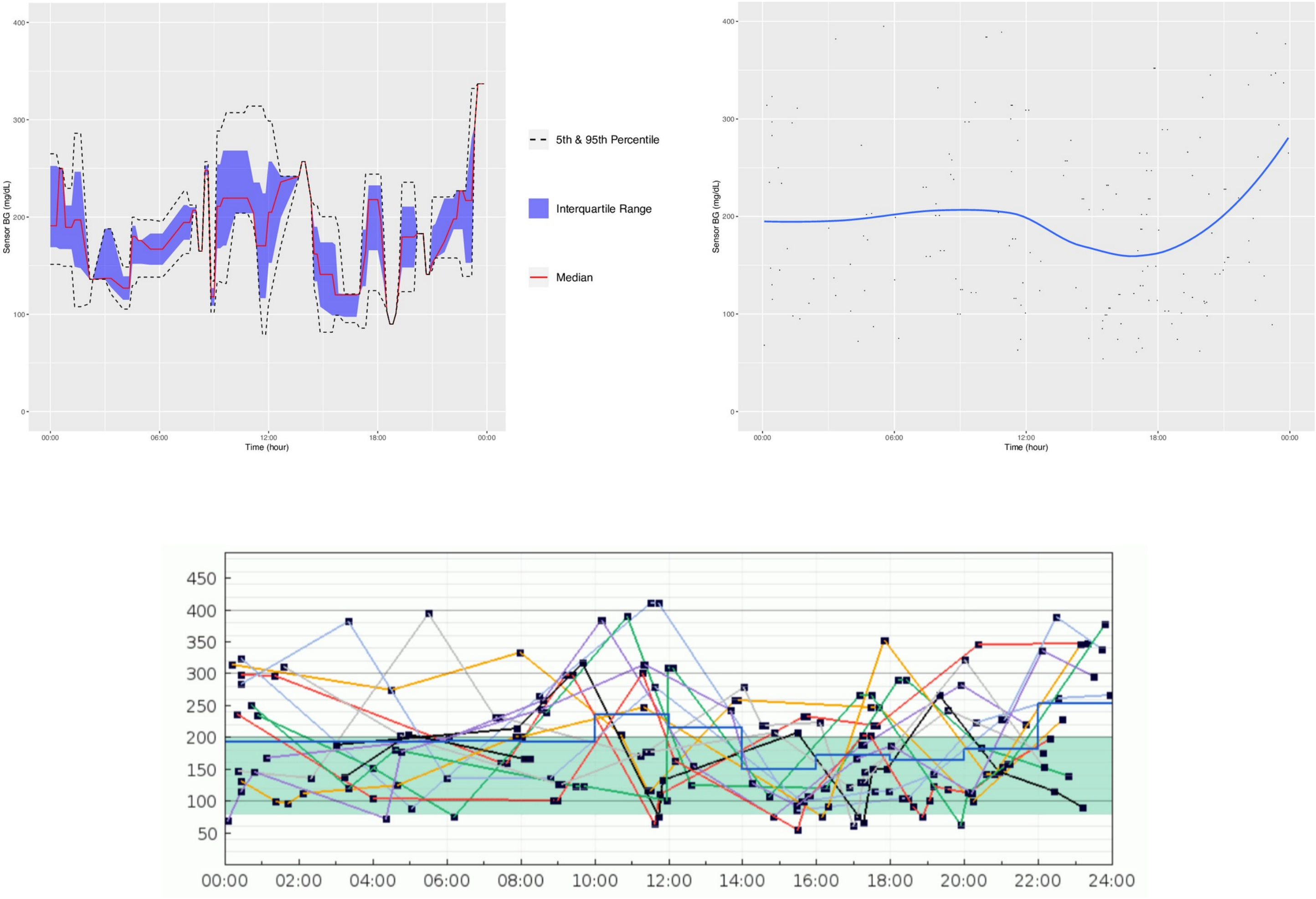

Fig 7. "cgmanalysis" Package Plots Compared to Diasend Daily Overlay Clockwise from top left: Aggregate Daily Overlay (Tukey Smoothing), Aggregate Daily Overlay (Loess Smoothing), Diasend Daily Overlay.

https://doi.org/10.1371/journal.pone.0216851.g007

Figs 4–7 show the comparisons of the graphical outputs produced by the proprietary software and the cgmanalysis package. In the graphs produced by the cgmanalysis package, glycemic patterns at each hour of the day are clearly visible and match the CGM device outputs well. However, some of the proprietary software appear to apply different smoothing algorithms, resulting in slightly different patterns across time.

Discussion

The summary variables produced by the cgmanalysis package match those from the proprietary software for all platforms assessed, and differences are mainly due to rounding discrepancies. Compared to the iPro 2, the number of high excursions differed by 1. Without access to the iPro algorithms we are unable to determine why these counts disagree, but the difference is not likely of clinical significance. The graphical outputs from the cgmanalysis package are similar to the CGM device output in terms of the glycemic patterns by hour of day, although there are small differences, likely due to different smoothing algorithms.

There are several limitations to our comparison of the cgmanalysis package to the proprietary software output. CGM devices only calculate a few summary variables, and accordingly it is difficult to test this package cohesively. Also, gold standard calculations do not exist for many of these variables, which makes verifying our results difficult. We hope that by making this package freely available and open source, these limitations will be minimized through widespread testing. Perhaps the greatest limitation to the software itself is the lack of an easy to use graphical user interface (GUI), which may prevent its use by clinicians with limited programming experience. We have included detailed documentation in the CRAN package, as well as a new-user guide on GitHub, but using the package still requires enough technical knowledge that it may be inaccessible to some users. None of the authors are software engineers, and the package is undoubtedly less efficient than it could be. Again, we hope that the free and open source nature will contribute significantly to improving the code over time, both as a result of outside contributions and our own planned updates.

In conclusion, our software provides a standardized, free, open-source approach to manage and analyze CGM data, enabling sharing of data across technology platforms, collaboration between research groups, and more effective use of the growing pool of CGM data. The advantage of using R functions rather than licensed statistical software, or a web-based or desktop application, is that R is freely available and open source. Clinicians or investigators can alter the code according to their needs and anyone can contribute to the development of the program, as CGM research and technology advance.

Author Contributions

Conceptualization: Tim Vigers, Gregory Forlenza, Laura Pyle.

Data curation: Tim Vigers.

Formal analysis: Tim Vigers.

Funding acquisition: Christine L. Chan, Janet Snell-Bergeon, Philip S. Zeitler, Gregory Forlenza.

Investigation: Tim Vigers, Petter Bjornstad, Gregory Forlenza, Laura Pyle.

Methodology: Tim Vigers, Gregory Forlenza, Laura Pyle.

Resources: Christine L. Chan, Janet Snell-Bergeon, Petter Bjornstad, Gregory Forlenza.

Software: Tim Vigers.

Supervision: Christine L. Chan, Janet Snell-Bergeon, Philip S. Zeitler, Gregory Forlenza, Laura Pyle.

Validation: Tim Vigers, Petter Bjornstad, Gregory Forlenza, Laura Pyle.

Visualization: Tim Vigers, Gregory Forlenza.

Writing – original draft: Tim Vigers, Laura Pyle.

Writing – review & editing: Tim Vigers, Christine L. Chan, Janet Snell-Bergeon, Petter Bjornstad, Philip S. Zeitler, Gregory Forlenza, Laura Pyle.

References

1. DeSalvo DJ, Miller KM, Hermann JM, Maahs DM, Hofer SE, Clements MA, et al. Continuous glucose monitoring and glycemic control among youth with type 1 diabetes: International comparison from the T1D Exchange and DPV Initiative. Pediatr Diabetes 2018; 19(7): 1271–1275. https://doi.org/10.1111/pedi.12711 PMID: 29923262
2. Beck RW, Bergenstal RM, Riddlesworth TD, Kollman C, Li Z, Brown AS, et al. Validation of Time in Range as an Outcome Measure for Diabetes Clinical Trials. Diabetes Care 2019; 42(3): 400–405. https://doi.org/10.2337/dc18-1444 PMID: 30352896

3. Bergenstal RM, Beck RW, Close KL, Grunberger G, Sacks DB, Kowalski A, et al. Glucose Management Indicator (GMI): A New Term for Estimating A1C From Continuous Glucose Monitoring. Diabetes Care 2018; 41(11): 2275–2280. https://doi.org/10.2337/dc18-1581 PMID: 30224348

4. Hernandez TL, Barbour LA. A standard approach to continuous glucose monitor data in pregnancy for the study of fetal growth and infant outcomes. Diabetes Technol Ther 2013; 15(2): 172–9. https://doi.org/10.1089/dia.2012.0223 PMID: 23268584

5. Hill NR, Oliver NS, Choudhary P, Levy JC, Hindmarsh P, Matthews DR. Normal reference range for mean tissue glucose and glycemic variability derived from continuous glucose monitoring for subjects without diabetes in different ethnic groups. Diabetes Technol Ther 2011; 13(9): 921–8. https://doi.org/10.1089/dia.2010.0247 PMID: 21714681

6. Sechterberger MK, Luijf YM, Devries JH. Poor agreement of computerized calculators for mean amplitude of glycemic excursions. Diabetes Technol Ther 2014; 16(2): 72–5. https://doi.org/10.1089/dia.2013.0138 PMID: 24191760

7. Zhang XD, Zhang Z, Wang D. CGManalyzer: an R package for analyzing continuous glucose monitoring studies. Bioinformatics 2018; 34(9): 1609–1611. https://doi.org/10.1093/bioinformatics/btx826 PMID: 29315360

8. Danne T, Nimri R, Battelino T, Bergenstal R, Close KL, DeVries JH, et al. International Consensus on Use of Continuous Glucose Monitoring. Diabetes Care 2017; 40(12): 1631–1640. https://doi.org/10.2337/dc17-1600 PMID: 29162583

9. Baghurst PA. Calculating the mean amplitude of glycemic excursion from continuous glucose monitoring data: an automated algorithm. Diabetes Technol Ther 2011; 13(3): 296–302. https://doi.org/10.1089/dia.2010.0090 PMID: 21291334

10. Harris PA, Taylor R, Thielke R, Payne J, Gonzalez N, Conde JG. Research electronic data capture (REDCap)—a metadata-driven methodology and workflow process for providing translational research informatics support. J Biomed Inform 2009; 42(2): 377–81. https://doi.org/10.1016/j.jbi.2008.08.010 PMID: 18929686

11. Tukey JW. Exploratory data analysis. 1st ed. Reading MA: Addison-Wesely; 1970.

12. Chambers JM, Hastie T. Statistical models in S. Boca Raton, FL: Chapman & Hall/CRC; 1992.

13. Wood SN. mgcv: GAMs and generalized ridge regression for R. R News 2001; 1(2): 20–25.

14. O'Sullivan F, Yandell BS, Raynor WJ. Automatic Smoothing of Regression Functions in Generalized Linear Models. J Am Stat Assoc 1986; 81(393): 96–103.