An Update on the iglu Software Package for Interpreting Continuous Glucose Monitoring Data

Elizabeth Chun[1], Nathaniel J Fernandes[2], Irina Gaynanova[3,*]

[1]Department of Statistics, Texas A&M University, College Station, TX, USA

[2]Department of Electrical and Computer Engineering, Texas A&M University, College Station, TX, USA

[3]Department of Biostatistics, University of Michigan, Ann Arbor, MI, USA

Short running title: Update of iglu software for CGM data

*Correspondence: Irina Gaynanova, irinagn@umich.edu

Keywords: CGM; diabetes mellitus; glycemic control; glycemic variability; postprandial excursions; software

Abstract

Background: Continuous glucose monitors (CGMs) are increasingly used to provide detailed quantification of glycemic control and glucose variability. An open-source R package iglu has been developed to assist with automatic CGM metrics computation and data visualization, providing a comprehensive list of implemented CGM metrics. Motivated by the recent international consensus statement on CGM metrics and recommendations from recent reviews of available CGM software, we present an updated version of iglu with improved accessibility and expanded functionality.

Methods: The functionality was expanded to include automated computation of hypo- and hyperglycemia episodes with corresponding visualizations, composite metrics of glycemic control (Glycemia Risk Index (GRI), Personal Glycemic State (PGS)), and glycemic metrics associated with postprandial excursions. The algorithm for Mean Amplitude of Glycemic Excursions (MAGE) has been updated for improved accuracy, and the corresponding visualization has been added. Automated hierarchical clustering capabilities have been added to facilitate statistical analysis. Accessibility was improved by providing support for the automatic processing of common data formats, expanding the graphical user interface, and providing mirrored functionality in Python.

Results: The updated version of iglu has been released to the Comprehensive R Archive Network (CRAN) as version 4. The corresponding Python wrapper has been released to the Python Package Index (PyPI) as version 1. The new functionality has been demonstrated using CGM data from 19 subjects with prediabetes and type 2 diabetes.

Conclusions: An updated version of iglu provides comprehensive and accessible software for analyses of CGM data that meets the needs of researchers with varying levels of programming experience. It is freely available on CRAN and on GitHub at https://github.com/irinagain/iglu.

1 Introduction

Continuous glucose monitors (CGMs) provide high-frequency measurements of interstitial glucose. They are increasingly used both in clinical practice and in research studies to provide detailed quantification of glycemic control and glucose variability. Multiple metrics for summarizing and interpreting CGM data have been proposed[1-2] covering various aspects of glycemic control such as overall glucose levels (e.g., mean), overall glucose variability (e.g., time in range, coefficient of variation), local glucose variability (e.g., MAGE,[3] the standard deviation of glucose rate of change[4]), hypoglycemic risk (e.g., Hypo Index), and hyperglycemic risk (e.g., Hyper Index).[5] New CGM metrics continue to be proposed and developed, including composite metrics that account for multiple dimensions of glucose control (e.g., PGS,[6] GRI,[7] COGI[8]). CGM data interpretation is often enhanced by graphical displays, such as a consensus bar chart of times in range that is part of the Ambulatory Glucose Profile (AGP) report or a histogram of glucose rates of change.[4] The variety of available CGM metrics and graphical display options poses continuous challenges for researchers and clinicians, as many metrics require nontrivial calculations. As a result, multiple software packages have been developed to assist with automatic CGM metrics computation and data visualization. Recent reviews and comparisons across packages are provided by Piersanti et al., 2023[9] and Olsen et al., 2024.[10]

An open-source R package `iglu`[11] has been developed to assist with automatic CGM metrics computation and data visualization, providing a comprehensive list of implemented CGM metrics, while also being both accessible to users with no programming experience (due to the graphical user interface via Shiny App at https://irinagain.shinyapps.io/shiny_iglu/) and convenient for quantitative researchers relying on reproducible scripts for all data processing and metric calculation steps. The combination of open-source environment and corresponding functionality made `iglu` attractive to a wide user base, leading to 18,000 downloads from the Comprehensive R Archive Network (CRAN) as of February 2024, and its recent application in CGMap project,[12] which provides reference values of CGM metrics based on CGM data from 7,000 non-diabetic individuals. However, as new CGM metrics continue to be proposed and developed, it becomes necessary to expand functionality over time. Motivated by the recent consensus statement on CGM metrics for clinical trials,[13] as well as recommendations for CGM software provided in recent reviews,[9-10] in this work, we present an updated version of `iglu` (relative to the original version[11]) with expanded functionality and enhanced accessibility.

Introduction of all new functionality is strictly guided by updated consensus definitions and recommendations[13], recommendations from most recent CGM software reviews[9, 10] or user requests (via direct emails to the authors or new created issues on the package GitHub website https://github.com/irinagain/iglu/issues). This functionality is not meant to replace the commonly adopted

characterization of CGM data via the Ambulatory Glucose Profile (AGP), already implemented in the prior version of iglu[11], but rather to complement this characterization[13]. The expanded functionality of `iglu` covers the quantification of hypo- and hyperglycemia episodes in accordance with consensus classification,[13] automated computation of composite metrics of glycemic control,[6-7] quantification of postprandial glucose excursions,[15] and the ability to separate metric calculations into day and night periods. While some of these features are also available in other CGM software platforms[9, 10], to the best of our knowledge iglu is the only software that provides automatic quantification of postprandial glucose excursions. Recently, this functionality has been successfully used to quantify the impact of sedentary behavior on postprandial glucose in older adults with overweight or obesity[15]. The algorithm for calculating the Mean Amplitude of Glycemic Excursions (MAGE) is replaced with a more accurate validated version.[3] Prompted by the identified lack of statistical methods implementation in CGM software,[10] the updated version also incorporates hierarchical clustering functionality with corresponding visualization, which can be used for both the identification of CGM metrics with complementary information as has been successfully demonstrated by CGMap project[12] as well as patient clustering. To improve accessibility, we address the limitations of the previous iglu version by providing support for specialized data formats from popular CGMs, expanding the graphical user interface, and releasing a mirror version of the software in the Python programming language tailored to computational researchers. We illustrate the functionality on public CGM data from 19 subjects with pre-diabetes and type 2 diabetes,[17] also made available as part of the update.

2 Methods

2.1 Metrics expansion

Episode calculation functionality has been added via the function `episode_calculation`, where episodes are defined as extended periods of hypo- or hyperglycemia, with a consecutive period of 15 minutes being used as a default minimal period in agreement with the consensus guidelines.[13] By default, the hypoglycemia thresholds are set at 70 mg/dL (level 1) and 54 mg/dL (level 2), and the hyperglycemia thresholds are set at 180 mg/dL (level 1) and 250 mg/dL (level 2). The function introduces 4 new metrics for each episode type (hypo- or hyperglycemia, level 1 or level 2): 1. average number of episodes per day, 2. average episode duration, 3. average glucose value during the episodes, and 4. total number of episodes. Level 2 episodes are nested within level 1 episodes, while exclusive level 1 episodes are indicated with the “excl” abbreviation in the returned variable names. Optional user customization of episode length and thresholds has also been added. An accompanying episode visualization coded by color has been implemented in the `epicalc_profile` function.

Two new composite metrics, GRI (Glycemic Risk Index)[7] and PGS (Personal Glycemic State)[6], have been added, with mathematical formulas taken from original publications. Three new metrics summarizing postprandial glucose excursions have also been added via the function `meal_metrics`: ΔG (the change in glucose from baseline to postprandial peak), ΔT (the time to postprandial peak from the start of the meal), and % Baseline Recovery (decrease in glucose one hour after postprandial glucose peak expressed as a % of the peak amplitude, i.e. ΔG).[15] We define baseline value as the average glucose value over an hour window before meal intake, with optional user customization of this time window available via `before_win` function argument. Figure 1 illustrates these metrics using an example postprandial excursion.

Finally, the functionality to separate metrics calculation into sleep and wake periods has been added via the function `calculate_sleep_wake`. By default, the sleep period is defined from midnight to 6 am, and the wake period is defined as the remaining hours. The user can adjust the default definition of sleep period.

To assist users whose primary interest is in the calculation of only consensus metrics[13], we have added `consensus_only` option to `all_metrics` function, which results in calculation of only a subset of metrics: percentage of sensor data obtained, time in ranges, mean, GMI, GRI, CV, SD, time in tight range (70-140 mg/dL), extended hypoglycemic and hyperglycemic event rates based on episode calculation functionality (these metrics are listed in Tables 2 and 3 of Battelino et al., 2023[13]).

2.2 MAGE calculation

The mean amplitude of glycemic excursions (MAGE) is originally defined by Service et al[17] as the arithmetic mean of the amplitude (height) of glucose excursions greater than the standard deviation of the glucose values. This definition led to a variety of algorithmic implementations since it requires automatic quantification of the visual “excursion” leading to substantial differences across software platforms in calculated values.[9,19] In the updated version of `iglu`, we implemented a recent algorithm for MAGE from Fernandes et al., 2022[3]. The algorithm uses the crosses of long and short-moving average glucose values to identify peaks and nadirs (defining excursions) automatically, with subsequent filtering only to select those excursions that are above the 1 standard deviation threshold. The optimal moving average window parameters have been developed based on readings of 5 min frequency[3]; and in the update, we internally interpolate to 5 min frequency, which leads to similar accuracy when the original frequency is 1 min, 10 min, or 15 min. To differentiate the new algorithm from the previous one, the acronym “ma” for the moving average is used in the specification of the MAGE version calculation (with the previous algorithm referred to as “naïve”). Since the excursions are not necessarily symmetric, the

algorithm separately calculates $MAGE_+$ and $MAGE_-$ corresponding to the averages of ascending and descending excursion amplitudes, respectively. The algorithm has been validated against manual MAGE calculations, achieving a correlation of 0.93, and shown to be more accurate than alternative implementations.[3] According to the original definition,[18] the final $MAGE_{Service}$ is either $MAGE_+$ or $MAGE_-$ depending on which amplitude was larger than 1 SD first. Other possible choices are $MAGE_{max}$ = max($MAGE_+$, $MAGE_-$) and $MAGE_{avg}$ = ($MAGE_+$ + $MAGE_-$)/2. We implemented the function `mage` that allows the calculation of all these variations, with $MAGE_{avg}$ being the default, as has been advocated by Baghurst et al., 2011.[20] The function automatically handles large missing data gaps. A more detailed description of the algorithm, together with additional validation results and cross-comparisons with existing MAGE algorithms, are available in the updated iglu MAGE vignette at https://irinagain.github.io/iglu/articles/MAGE.html.

2.3 Clustering and visualization

Clustering functionality has been added, allowing investigation of differences and similarities both across subjects and across glucose metrics via the function `metrics_heatmap`. Such clustering has been used in the original `iglu` paper[11] but has not been made available as a default functionality. Recently, a similar clustering analysis has been performed using `iglu` calculated metrics,[12] but relied on customized scripts. By default, the analysis is performed using all metrics implemented in `iglu`, but the user can customize this choice. Each selected metric is centered and scaled across subjects before the analyses. Hierarchical clustering is performed separately across metrics and across subjects with complete linkage as a default agglomeration method for each.[21] The dissimilarities are measured by the correlation distance, which is defined as $d(i, j) = 1 - r_{ij}$, where r_{ij} is the Pearson correlation between i_{th} and j_{th} metric (or subject). Both the choice of agglomeration method and the dissimilarity measures can be adjusted by the user. The resulting clustering dendrograms can be used to interpret aspects of glycemic control that are measured by distinct metric groups (clusters) and identify groups of subjects with distinct glycemic profiles. An accompanying visualization of the clusters is implemented using a heatmap with rows corresponding to metrics and columns to subjects. The function output contains two objects describing the tree produced by the clustering process, with `tree_col` corresponding to subjects and `tree_row` corresponding to metrics. These objects can be used to extract cluster assignments for subsequent data exploration and analysis purposes using `cutree` function.

2.4 Improved accessibility

Several enhancements were made to facilitate the use of `iglu` by a broad range of users. First, the `read_raw_data` function was added to enable the automatic processing of specialized data formats from popular CGMs into a unified format used by `iglu`. The underlying interface closely mimics similar functionality in R package cgmanalysis,[21] providing support for Dexcom, Libre, Libre Pro, and iPro data export formats.

Second, to increase accessibility for researchers with limited to no programming experience, we updated the `iglu` graphical user interface to incorporate the updated functionality, and also provided a pre-loaded example dataset to facilitate exploration of the interface. The pre-loaded dataset contains Dexcom G4 measurements from 5 subjects with type 2 diabetes and is also included in the original version of `iglu`.[11] The graphical user interface can be accessed locally in R using the iglu shiny() function and is also made available as a web-based interface at https://irinagain.shinyapps.io/shiny iglu/.

Finally, to increase accessibility for computational and machine learning researchers who are more likely to be familiar with Python rather than the R programming language, we released a Python package `iglu-py` that enables calling `iglu` R functions within the native Python interface without requiring manual R installation. The graphical user interface capability has also been made available in `iglu-py`. The Python version `iglu-py` can be installed from the standard Python Package Index (PyPI) repository package, and the development version is also available at https://github.com/IrinaStatsLab/iglu-py.

2.5 Testing and validation

To test and validate the updated `iglu` functionality, particularly when calculating metrics associated with postprandial glucose excursions, a new example dataset was added. This dataset contains Dexcom G4 measurements from 19 subjects that are part of the larger study from Hall et al., 2018.[16] The purpose of the original study was to evaluate how postprandial glucose patterns differ across individuals given a standardized nutrient challenge. The selected 19 subjects were diagnosed with either type 2 diabetes (5 subjects) or pre-diabetes (14 subjects) on screening tests. In addition to CGM data, we also included data associated with standardized meal intakes, including the timestamp associated with the start of the meal and the meal type. All standardized meals were taken at breakfast and corresponded to one of the three meal types: cornflakes and milk (CF), peanut butter sandwich (PB), or PROBAR protein bar (Bar). Subject 2133-039, diagnosed with type 2 diabetes, was used to illustrate episode functionality. Subjects 1636-69-001 (type 2 diabetes), 1636-69-026 (pre-diabetes), and 1636-69-032 (pre-diabetes) were used for PGS, GRI, and separate calculations across sleep and wake periods. Subject 2133-018 (type 2 diabetes), with available data on three meals corresponding to CF, PB, and PROBAR, was used for meal-

related metrics. Subject 2133-019 (pre-diabetes) was used for the MAGE algorithm. All 19 subjects were used for clustering and verification of Python capabilities, with the first 5 subject IDs being used to illustrate the latter.

3. Results

The new version of `iglu` includes a comprehensive list of available glucose metrics with the addition of clustering functionality and improved accessibility. Table 1 summarizes new functionalities relative to the previous versions. The new functionality was verified using a newly added CGM dataset from Hall et al., 2018[16], available as the `example_data_hall` object.

3.1. Metrics expansion

To illustrate episode calculation functionality, we use subject 2133-039, diagnosed with type 2 diabetes, as an example. Figure 2 illustrates the output of `episode_calculation` function and the companion episode visualization functionality implemented in `epicalc_profile`. The table included in the episode visualization matches the code output of the episode calculation function. The graphical display, color-coded by level and episode type, allows visual inspection of episode patterns and validation of results presented in the table. There are 10 hypoglycemia episodes, corresponding to 1.3 episodes per day. In contrast, there are only 2 hyperglycemia episodes, both strictly within level 1 (above 180 mg/dL but below the level 2 threshold of 250 mg/dL).

The values of two new composite metrics, GRI and PGS, were evaluated on three selected subjects using corresponding functions.

```
> gri(example_data_hall)
# A tibble: 3 x 2
  id              GRI
  <chr>         <dbl>
1 1636-69-001   3.34
2 1636-69-026   0.624
3 1636-69-032   0.269

> pgs(example_data_hall)
# A tibble: 3 × 2
```

```
id             PGS
  <chr>        <dbl>
1 1636-69-001   5.51
2 1636-69-026   5.44
3 1636-69-032   5.32
```

PGS values are comparable for all subjects, whereas GRI is notably larger for subjects 1636-69-001. Based on the diagnosis from the screening tests, this subject has type 2 diabetes. In contrast, the other two subjects have pre-diabetes, which is reflected by the lower GRI values. These differences in PGS are less pronounced due to the difference in the constituent components for each composite metric. While GRI considers exclusively the percent of time spent out of range, PGS also incorporates information about hypoglycemic episodes, as well as glucose variability metrics.

Functionality associated with meal-related metrics has been validated using subject 2133-018, diagnosed with type 2 diabetes, with available data on three meals corresponding to peanut butter (PB), cornflakes (CF), and PROBAR (Bar) meal types, respectively. Figure 3 illustrates the matched CGM data visualization with the corresponding mealtimes and output of `meal_metrics` function. The function automatically matches CGM data with the supplied mealtime information to compute three metrics: ΔG, ΔT, and % Baseline Recovery. For all three meals, the glucose peak is reached approximately 80 minutes (ΔT) after the meal's start time. The CF meal type leads to the highest glucose excursion, with an increase of 171 mg/dL (ΔG) compared to the pre-meal baseline, whereas the glucose increase from Bar intake is the smallest (ΔG = 78 mg/dL). The subject displays relatively poor recovery from the meal intakes, reaching only 38% recovery one hour after peak for PB meal type. The recovery is best for the Bar meal type, which could be due to its smallest corresponding ΔG.

To validate new functionality associated with calculating metrics separately across sleep and wake periods, we compute mean glucose for each period for three selected subjects as illustrated below.

```
> calculate_sleep_wake(example_data_hall, mean_glu, calculate = "both")
# A tibble: 3 × 3
   id           `mean sleep` `mean wake`
   <chr>               <dbl>       <dbl>
 1 1636-69-001          99.0       111.
 2 1636-69-026         113.        116.
 3 1636-69-032         103.        110.
```

We find that the average glucose values are higher during the wake period compared to the sleep period (defined by default as midnight to 6 am), which is expected since high glucose values tend to be associated with meal intakes, which happen during the day.

Finally, we validated that the new `consensus_only` option in `all_metrics` returns a reduced list of metric values that agree with the values from individual metric function calls. Below is an example of the function use based on CGM data from subject 2133-018.

```
> all_metrics(data = example_data_hall %>% dplyr::filter(id == "2133-039"),
                  metrics_to_include = 'consensus_only')
```

3.2 MAGE Calculation

The new MAGE algorithm has been implemented with corresponding visualization to allow the users to examine the identified excursions and visualize any data gaps. Using data from subject 2133-019 as an example, the MAGE value is computed by the call to `mage_ma_single` function, where “ma” in the function name stands for moving average to differentiate the new algorithm from other MAGE implementations. The optional visualization is available via `plot = TRUE`, allowing the user to examine underlying calculations (Figure 4). In this instance, the output warns of a large missing data gap. The illustration reveals two missing data gaps (highlighted by shaded regions), with one of the gaps spanning several days of measurements. In the presence of large gaps (over 12 hours of missing data), excursion identification is performed separately for each consecutive segment, with the final MAGE value returned as a weighted average across segments (adjusting for segment length). Each segment’s start and end boundaries are shown with solid and dotted vertical lines, respectively. The arrows indicate the direction of calculation ($MAGE_+$ or $MAGE_-$), with both arrows shown as default (corresponding to $MAGE_{avg}$). The up arrows start from the automatically identified nadir, with the arrow height corresponding to the height of the peak, and the down arrows start from the peak. By further setting plot `type = ″plotly″`,[23] the user gets access to interactive capabilities, for example, the ability to hover over any datapoint to see an informative tooltip.

3.3 Clustering and visualization

To validate the new clustering functionality, we applied the metrics heatmap function to 19 subjects from Hall et al., 2018[17] and 57 extracted glucose metrics. Figure 5 illustrates the corresponding heatmap and clustering dendrograms, where, by default, the metrics are visually divided into 6 clusters. The accompanying subjects’ dendrogram visually separates the subjects into three groups, which we would interpret from left to right. The 1st group consists of subjects with higher “in-range” values (higher values for 1st metric group) and lower values for hypo- and hyperglycemia metrics. The 2nd group consists of

subjects with more pronounced hypoglycemia. The 3rd group comprises subjects with more pronounced hyperglycemia and higher glucose variability. By matching the subjects with the diagnosis, we observe that the first two groups primarily correspond to pre-diabetes patients. In contrast, the last group has an equal mix of patients with pre-diabetes and type 2 diabetes. To automatically extract subjects' cluster assignments, we use the returned `tree_col` object with `cutree` function:

```
> cluster_out = metrics_heatmap(data = example_data_hall)
> cutree(cluster_out$tree_col, k = 3)
 1636-69-001  1636-69-026  1636-69-032  1636-69-090  1636-69-091  1636-69-114
           1            2            2            1            2            2
1636-70-1005 1636-70-1010     2133-004     2133-015     2133-017     2133-018
           1            3            1            2            2            1
    2133-019     2133-021     2133-024     2133-027     2133-035     2133-036
           3            1            3            3            2            3
    2133-039
           3
```

This clustering information can be used for subsequent data exploration and analysis.

2.4 Improved accessibility

The updated `iglu` functionality has been incorporated into a point-and-click graphical user interface (GUI) via the Shiny app, which is available from the R console by calling:

```
> iglu::iglu_shiny()
```

or directly at https://irinagain.shinyapps.io/shiny iglu/. The pre-loaded data on 5 subjects with type 2 diabetes allows the users to get accustomed to the functionality quickly. Figure 6 illustrates the newly added episode calculation interface, with dedicated boxes for adjustment of default parameters corresponding to level 1 and level 2 hypo- and hyperglycemia thresholds and minimal duration length (in minutes) to be counted as an episode. This functionality is added as a new tab in addition to the already existing AGP report tab[14], providing the users with complementary information.

To validate the functionality of the mirrored Python package, we installed it using the Python package manager via:

```
$ pip install iglu-py
```

We verified `iglu` functionality within the Python environment by calculating average glucose values for data from Hall et al.[17] The corresponding script in Python is:

```
>>> import iglu_py
>>> iglu_py.mean_glu(iglu.example_data_hall)
        id           mean
1   1636-69-001   108.228602
2   1636-69-026   115.155902
3   1636-69-032   108.315760
4   1636-69-090   108.750403
5   1636-69-091   103.107044
```

and the corresponding output is verified to match the R output. The graphical user interface is also accessible from the Python environment via the following command.

```
>>> iglu_py.iglu_shiny()
```

4 Conclusion

We provide an updated version of `iglu` with the intent of meeting the needs for automatic calculation of a wide array of CGM metrics that continue to be introduced in the literature, and to improve the accessibility of software for both clinical researchers (via an expanded graphical user interface) and computational researchers (via the addition of Python interface). Our update also addresses limitations in the original `iglu` version[11] by including functionality to read data in specialized formats of many popular CGMs, quantify hypo- and hyperglycemic episodes, and separate it into sleep and wake periods. We have demonstrated the updated functionality on the CGM data from patients with prediabetes and type 2 diabetes,[17] and these data are also available with the software.

Many alternative CGM software platforms are available, such as EasyGV,[24] GlyCulator,[25] cgmanalysis,[26] AGATA,[27] etc., which provide complementary solutions to the needs of various research groups. A detailed comparison across CGM software packages, including comparison with `iglu`, is provided in recent comprehensive software reviews.[9-10] The main strength of `iglu` is its comprehensive implementation of available CGM metrics and graphical displays,[9-10] as well as demonstrated agreement in calculated metric values against other CGM software.[9] Of particular note is the new MAGE algorithm, which has been validated against manual calculations,[3] showing higher accuracy than several existing alternatives. To further aid users in ensuring the accuracy of results, we have added customized graphical displays (e.g., new episode calculation display in Figure 2 and a new display for MAGE in Figure 3) to enable visual inspection of underlying calculations.

Objective comparisons of different CGM software packages pose multiple challenges. Usability and ease of use assessments can differ greatly among user groups. For example, quantitative researchers might prefer tools that facilitate reproducible research in familiar programming environments, while users with limited programming skills may favor a user-friendly graphical interface. Moreover, a thorough comparison of performance metrics like accuracy and speed necessitates extensive testing across diverse datasets and continuous software updates, which are beyond the scope of our current study. Recent efforts such as those by Piersanti et al. (2023)[9], which evaluates and compares a comprehensive list of metrics and software platforms based on CGM data from 6 patients with type 1 diabetes, and Akturk et al. (2022)[26], which considers a larger cohort of 188 patients with type 1 diabetes but limits its comparison to specific software and only few summary metrics, illustrate the limitations of current methodologies. These examples highlight the critical need for more comprehensive cross-evaluations and the establishment of universal standards for comparing software packages in future studies.

As with any CGM software, the accuracy of calculated metrics can be affected by missing data. By default, `iglu` performs linear interpolation in the presence of short missing data gaps (using a 45 min threshold as default), and provides warnings to the user whenever a large gap (over 12 hours) is present in the data. However, linear interpolation is only applied for the calculation of more complex metrics that require equally-spaced measurements (e.g., the new MAGE algorithm, episode calculation functionality) and is not automatically applied when calculating mean glucose values and time-in-range. While other CGM software typically use a similar approach (e.g., 50 min gap is used by default in EasyGV[24]), large amounts of missing data may lead to disagreements in calculated metric values depending on the chosen mechanism of handling missing data. Extra care should be taken when data is not missing at random but is due to the values outside of thresholds corresponding to CGM measurement range, which is dependent on the specific CGM model. By default, `iglu` treats such values the same as other missing data, whereas other software programs may choose to replace those values with values at the threshold. While the former approach avoids the downward bias in glucose variability metrics resulting from imputing a constant threshold limit for missing values, the latter approach can be more accurate for evaluating time-in-range measures. However, the latter choice requires explicit knowledge of the measurement limits associated with a specific CGM model and thus may be error-prone when applied on a large scale with data from several different manufacturers. A preferred solution would be to perform sensitivity analyses of calculated metric values depending on the choice of missing data treatment.[24] However, such customized analyses are difficult to implement in an automated fashion.

Many CGM metrics are highly correlated[28, 29, 30], raising the question of the added value of having a comprehensive array of metrics over the consensus ones. Large-scale prospective outcome studies

would be valuable for examining metrics overlap and their predictive power for clinical outcomes[1]. By providing accessible software that allows computation of the wide array of CGM metrics in a reproducible fashion, we hope to facilitate future research in this direction.

In summary, we believe that an updated version of `iglu` provides comprehensive and accessible software for analyses of CGM data. It accommodates both researchers who favor point-and-click graphical user interfaces as well as those who prefer the creation of reproducible scripts for all data processing and metric calculation steps.

Authors' Contributions

E.C.: investigation; software; validation; writing - original draft; N.F: investigation; software; validation; writing - original draft; I.G.: conceptualization; investigation; data curation; supervision; writing – original draft; writing - review and editing.

Funding Information

This research was supported by NSF DMS 2044823.

Author Disclosure Statement

No competing financial interests exist.

References

1. Gaynanova I. Digital Biomarkers of Glucose Control - Reproducibility Challenges and Opportunities. Biopharmaceutical Report. 2022;29(1):21-6.

2. Nguyen M, Han J, Spanakis EK, Kovatchev BP, Klonoff DC. A Review of Continuous Glucose Monitoring-Based Composite Metrics for Glycemic Control. Diabetes Technology & Therapeutics. 2020;22(8):613-22.

3. Fernandes NJ, Nguyen N, Chun E, Punjabi NM, Gaynanova I. Open-Source Algorithm to Calculate Mean Amplitude of Glycemic Excursions Using Short and Long Moving Averages. Journal of Diabetes Science and Technology. 2022 Mar;16(2):576-7.

4. Clarke W, Kovatchev BP. Statistical Tools to Analyze Continuous Glucose Monitor Data. Diabetes Technology & Therapeutics. 2009;11(S1):S45 S54.

5. Rodbard D. Interpretation of Continuous Glucose Monitoring Data: Glycemic Variability and Quality of Glycemic Control. Diabetes Technology & Therapeutics. 2009;11(s1):S-55 S-67.

6. Hirsch IB, Balo AK, Sayer K, Garcia A, Buckingham BA, Peyser TA. A Simple Composite Metric for the Assessment of Glycemic Status from Continuous Glucose Monitoring Data: Implications for Clinical Practice and the Artificial Pancreas. Diabetes Technology & Therapeutics. 2017 Jun;19(S3):S-38.

7. Klonoff DC, Wang J, Rodbard D, Kohn MA, Li C, Liepmann D, et al. A glycemia risk index (GRI) of hypoglycemia and hyperglycemia for continuous glucose monitoring validated by clinician ratings. Journal of Diabetes Science and Technology. 2023;17(5):1226-42.

8. Leelarathna L, Thabit H, Wilinska ME, Bally L, Mader JK, Pieber TR, et al. Evaluating Glucose Control With a Novel Composite Continuous Glucose Monitoring Index. Journal of Diabetes Science and Technology. 2019;14(2):277-83.

9. Piersanti A, Giurato F, G¨obl C, Burattini L, Tura A, Morettini M. Software Packages and Tools for the Analysis of Continuous Glucose Monitoring Data. Diabetes Technology & Therapeutics. 2023 Jan;25(1):69-85.

10. Olsen MT, Klarskov CK, Dungu AM, Hansen KB, Pedersen-Bjergaard U, Kristensen PL. Statistical Packages and Algorithms for the Analysis of Continuous Glucose Monitoring Data: A Systematic Review. Journal of Diabetes Science and Technology. 2024 Jan:19322968231221803.

11. Broll S, Urbanek J, Buchanan D, Chun E, Muschelli J, Punjabi NM, et al. Interpreting Blood GLUcose Data with R Package Iglu. PLoS ONE. 2021 Apr;16(4):e0248560.

12. Keshet A, Shilo S, Godneva A, Talmor-Barkan Y, Aviv Y, Segal E, et al. CGMap: Characterizing Continuous Glucose Monitor Data in Thousands of Non-Diabetic Individuals. Cell Metabolism. 2023 May;35(5):758-69.e3.

13. Battelino T, Alexander CM, Amiel SA, Arreaza-Rubin G, Beck RW, Bergenstal RM, et al. Continuous Glucose Monitoring and Metrics for Clinical Trials: An International Consensus Statement. The Lancet Diabetes & Endocrinology. 2023 Jan;11(1):42-57.

14. Johnson ML, Martens TW, Criego AB, Carlson AL, Simonson GD, Bergenstal RM. Utilizing the ambulatory glucose profile to standardize and implement continuous glucose monitoring in clinical practice. Diabetes technology & therapeutics. 2019; 21(S2), S2–S17. pmid:31169432

15. Service FJ. Glucose variability. Diabetes. 2013;62(5):1398-404.

16. Chun E, Gaynanova I, Melanson EL, Lyden K. Pre-Versus Postmeal Sedentary Duration—Impact on Postprandial Glucose in Older Adults With Overweight or Obesity. Journal for the Measurement of Physical Behaviour. 2024 Mar 27;7(1).

17. Hall H, Perelman D, Breschi A, Limcaoco P, Kellogg R, McLaughlin T, et al. Glucotypes Reveal New Patterns of Glucose Dysregulation. PLOS Biology. 2018 Jul;16(7):e2005143.

18. Service FJ, Molnar GD, Rosevear JW, Ackerman E, Gatewood LC, Taylor WF. Mean amplitude of glycemic excursions, a measure of diabetic instability. Diabetes. 1970;19(9):644-55.

19. Sechterberger MK, Luijf YM, DeVries JH. Poor Agreement of Computerized Calculators for Mean Amplitude of Glycemic Excursions. Diabetes Technology & Therapeutics. 2014;16(2):72-5.

20. Baghurst PA. Calculating the Mean Amplitude of Glycemic Excursion from Continuous Glucose Monitoring Data: An Automated Algorithm. Diabetes Technology & Therapeutics. 2011;13(3):296-302.

21. Murtagh F, Contreras P. Algorithms for Hierarchical Clustering: An Overview. WIREs Data Mining and Knowledge Discovery. 2012;2(1):86-97.

22. Vigers T, Chan CL, Snell-Bergeon J, Bjornstad P, Zeitler PS, Forlenza G, et al. cgmanalysis: an R package for descriptive analysis of continuous glucose monitor data. PLoS One. 2019;14(10):e0216851.

23. Sievert C. Interactive Web-Based Data Visualization with R, plotly, and shiny. Chapman and Hall/CRC; 2020.

24. Moscardó V, Giménez M, Oliver N, Hill NR. Updated Software for Automated Assessment of Glucose Variability and Quality of Glycemic Control in Diabetes. Diabetes Technology & Therapeutics. 2020 Oct;22(10):701-8.

25. Chrzanowski J, Grabia S, Michalak A, Wielgus A, Wykrota J, Mianowska B, et al. GlyCulator 3.0: A Fast, Easy-to-Use Analytical Tool for CGM Data Analysis, Aggregation, Center Benchmarking, and Data Sharing. Diabetes Care. 2023 Jan;46(1):e3-5.

26. Akturk HK, Vigers T, Forlenza G, Champakanath A, Pyle L. Comparison of Cgmanalysis, a Free Open-Source Continuous Glucose Monitoring Data Management and Analysis Software, with Commercially Available CGM Platforms: Data Standardization for Diabetes Technology Research. Diabetes Technology & Therapeutics. 2022 Jan;24(1):54-60.

27. Cappon G, Sparacino G, Andrea Facchinetti. AGATA: A Toolbox for Automated Glucose Data Analysis. Journal of Diabetes Science and Technology. 2023 Jan:19322968221147570.

28. Rodbard D. New and Improved Methods to Characterize Glycemic Variability Using Continuous Glucose Monitoring. Diabetes Technology & Therapeutics. 2009;11(9):551-65

29. Fabris C, Facchinetti A, Sparacino G, Zanon M, Guerra S, Maran A, et al. Glucose Variability Indices in Type 1 Diabetes: Parsimonious Set of Indices Revealed by Sparse Principal Component Analysis. Diabetes Technology & Therapeutics. 2014;16(10):644-52.

30. Fabris C, Facchinetti A, Fico G, Sambo F, Arredondo MT, Cobelli C, et al. Parsimonious Description of Glucose Variability in Type 2 Diabetes by Sparse Principal Component Analysis. Journal of Diabetes Science and Technology. 2015;10(1):119-24.

		Manuscript			
		Broll et al., 2021[11]		This paper	
		CRAN R package version			
		V1	V2	V3	V4
Metrics	Episode calculations	NA	NA	New feature	+ update to match consensus definition
	Postprandial (meal) metrics	NA	NA	NA	New feature
	Other	ADRR, BGIs, CONGA, CV, ea1c, GRADEs, hyper/hypo-index, IGC, J-index, linear interpolation, mean, M-value, MODD, SD measures, times in ranges	+ AUC, COGI, CV measures, GMI, GVP, MAD, MAG, rate of change, percent of sensor data, `all_metrics` wrapper	+ sleep/wake metric splitting	+ PGS, GRI, `consensus_only`[13] option in `all_metrics` to return reduced metrics list
MAGE Algorithm		NA	Naive algorithm (mean glucose values further than 1SD from the day's mean)	New algorithm[3] to match Service definition	+ updates to account for large gaps of missing data (> 3 hours)
Clustering and Visualizations		Time-series, lasagna plots, rates of change plot	+ AGP, day by day plots, bar plot for times in range	+ interactive MAGE plot, episode visualization	+ metrics heatmap, hierarchical clustering, meals plot
Accessibility			GUI via Shiny App, example dataset	+ support of specialized data formats; mmol/l to mg/dl conversion in `process_data`; example dataset in Shiny app	+ updated GUI to match R package; mirrored Python package

Table 1: A summary of iglu functionalities relative to released CRAN R package versions. The original manuscript[11] covers versions 1 and 2, with the current manuscript describing changes since then.

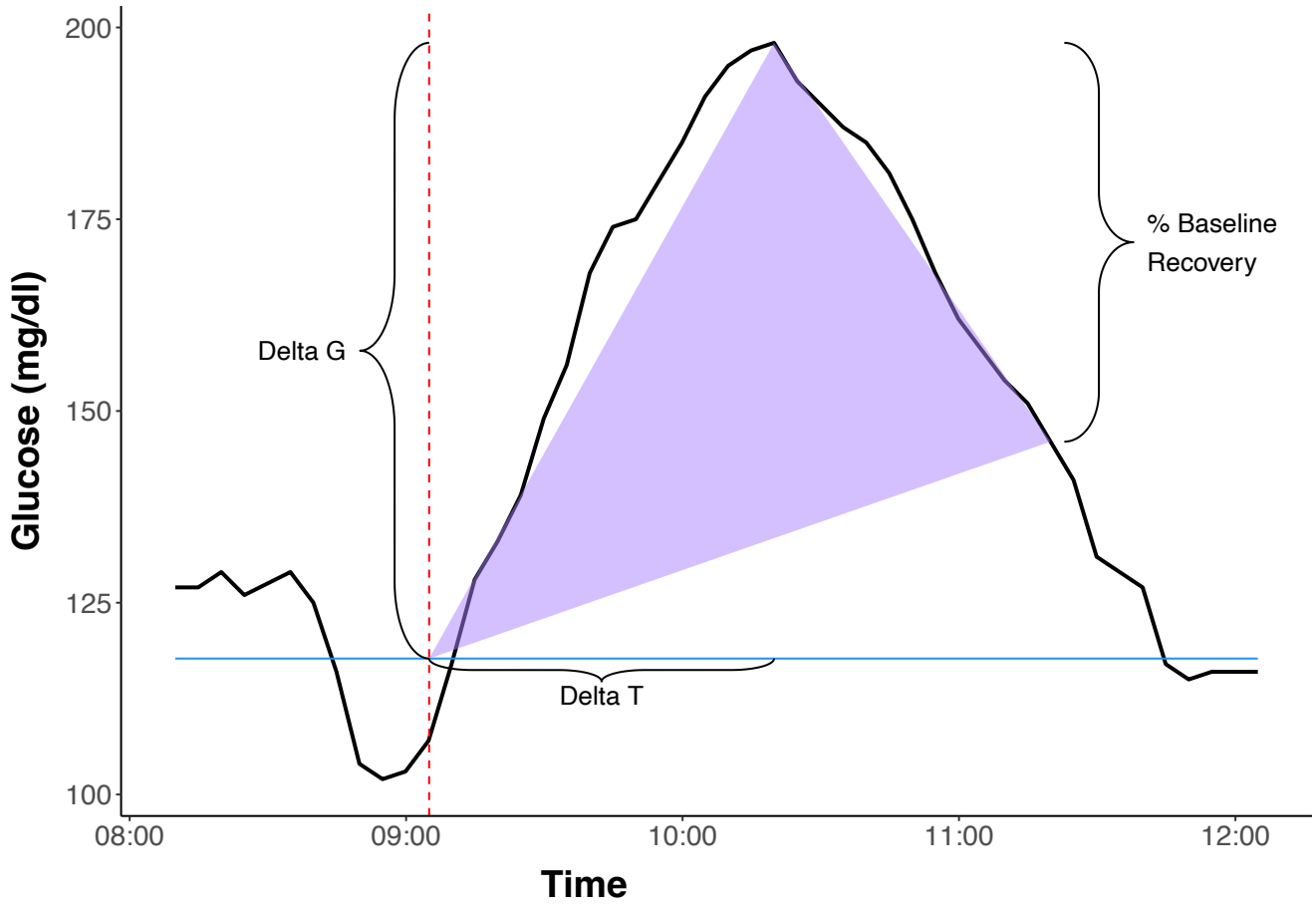


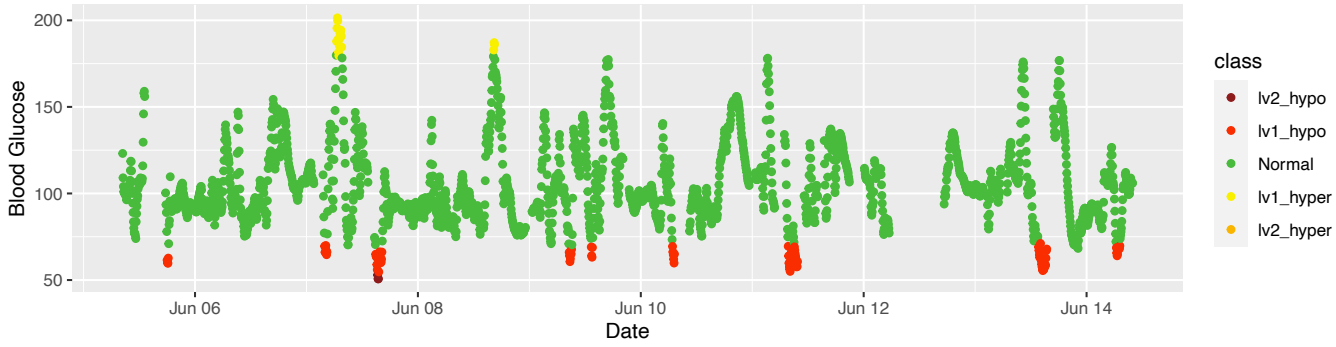
Figure 1: Illustration of metrics summarizing postprandial glucose excursion: ΔG (the change in glucose from baseline to postprandial peak), ΔT (the time to postprandial peak from the start of the meal), and % Baseline Recovery (decrease in glucose one hour after postprandial glucose peak expressed as a % of the peak amplitude, i.e. ΔG).

A

Episode Metrics – 2133–039

	Hypoglycemia Level 1	Hypoglycemia Level 2	Hypoglycemia Extended	Hyperglycemia Level 1	Hyperglycemia Level 2	Hypoglycemia Level 1 excl	Hyperglycemia Level 1 excl
Thresholds	<70 mg/dL	<54 mg/dL	<70 mg/dL	>180 mg/dL	>250 mg/dL	70–54 mg/dL	180–250 mg/dL
Avg Episodes/Day	1.33	0.13	0.00	0.27	0.00	1.20	0.27
Mean duration	49.00 min	15.00 min	0.00 min	42.50 min	0.00 min	45.00 min	42.50 min
Mean glucose	63.79 mg/dl	51.43 mg/dl	NA mg/dl	187.87 mg/dl	NA mg/dl	64.22 mg/dl	187.87 mg/dl
Total episodes	10.00	1.00	0.00	2.00	0.00	9.00	2.00

An episode is >= 15 continuous minutes

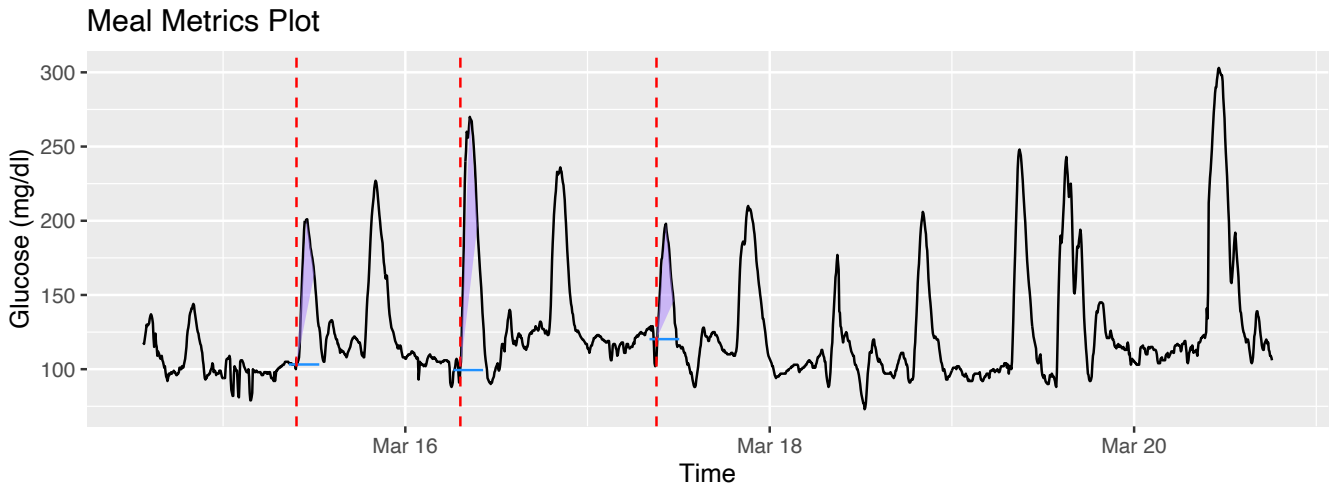


B

```
> episode_calculation(data = example_data_hall %>% dplyr::filter(id == "2133-039"))

# A tibble: 7 × 7

  id          type   level       avg_ep_per_day avg_ep_duration avg_ep_gl total_episodes

  <chr>     <chr> <chr>             <dbl>           <dbl>     <dbl>          <dbl>

1 2133-039 hypo  lv1                1.33             49        63.8             10

2 2133-039 hypo  lv2                0.133            15        51.4              1

3 2133-039 hypo  extended           0                 0        NA                0

4 2133-039 hyper lv1                0.266            42.5     188.               2

5 2133-039 hyper lv2                0                 0        NA                0

6 2133-039 hypo  lv1_excl           1.20             45        64.2              9

7 2133-039 hyper lv1_excl           0.266            42.5     188.               2
```

Figure 2: Illustration of episode visualization (Panel A) and corresponding console calculation (Panel B) functionality implemented in iglu using subject 2133-039 from Hall et al., 2018[17] as an example.

A



B

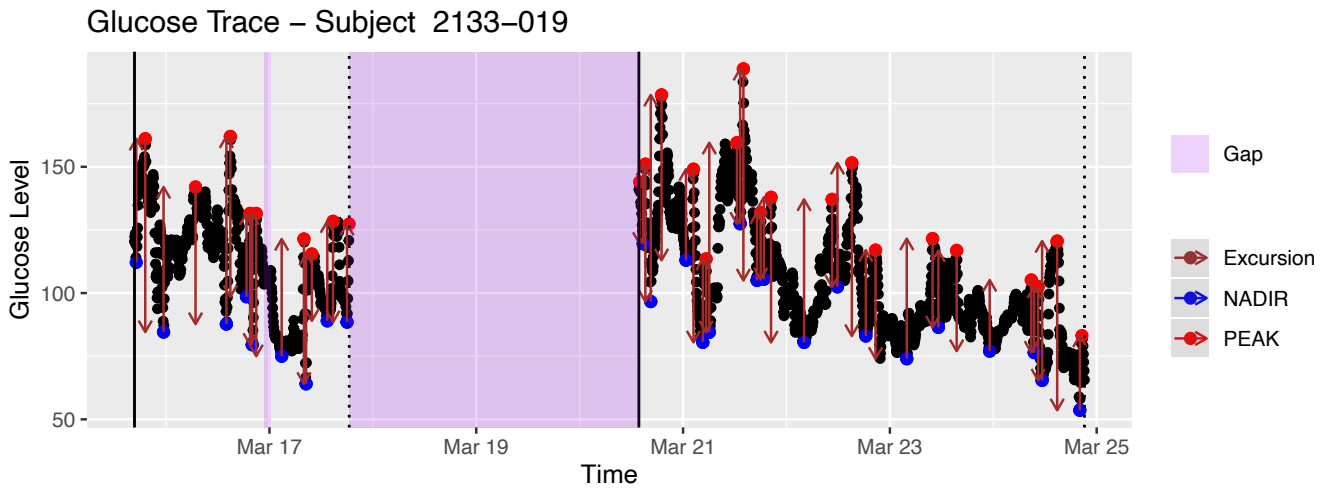
```
> example_meals_hall%>% dplyr::filter(id == "2133-018")

# A tibble: 3 × 3

  id       meal  mealtime

  <chr>    <chr> <dttm>

1 2133-018 PB 1  2017-03-15 09:40:00

2 2133-018 CF 1  2017-03-16 07:15:00

3 2133-018 Bar 1 2017-03-17 09:05:00
```

C

```
> meal_metrics(example_data_hall %>% dplyr::filter(id == "2133-018"),
                example_meals_hall%>% dplyr::filter(id == "2133-018"))

# A tibble: 3 × 6

  id       time                meal  deltag deltat basereco

  <chr>    <dttm>              <chr>  <dbl>  <dbl>    <dbl>

1 2133-018 2017-03-15 09:40:00 PB 1    97.8     80    0.378

2 2133-018 2017-03-16 07:15:00 CF 1   171.      75    0.429

3 2133-018 2017-03-17 09:05:00 Bar 1   77.6     75    0.670
```

Figure 3: Illustration of meal metrics visualization (Panel A), meal times in original data (Panel B), and corresponding metric values (Panel C) using subject 2133-039 from Hall et al., 2018[17] as an example.

A



B

```
> mage_ma_single(data = example_data_hall %>% dplyr::filter(id == "2133-019"), plot=FALSE)

Gap found in data for subject id: 2133-019, that exceeds 12 hours.

[1] 47.03813
```

Figure 4: Illustration of MAGE calculation (Panel B) and associated visualization (Panel A) in iglu using subject 2133-019 from Hall et al., 2018[17] as an example. The software automatically highlights regions with missing glucose readings (gaps). The colored dots correspond to automatically identified peaks and nadirs corresponding to excursions. The arrows indicate excursion magnitudes with arrows pointing up for $MAGE_{+}$ and pointing down for $MAGE_{-}$.

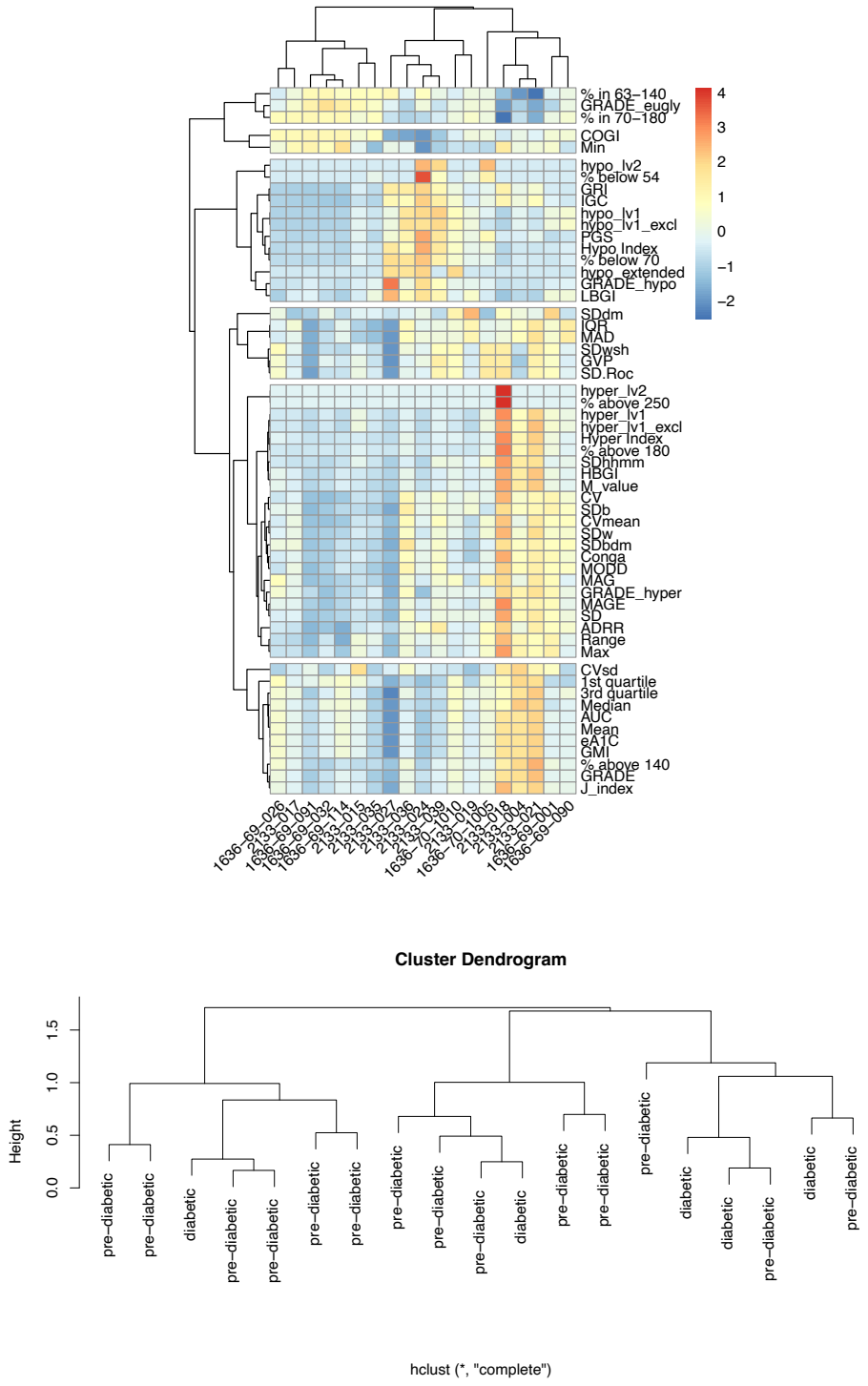


Figure 5: Top: Heatmap of CGM metrics on 19 subjects from Hall dataset[17]. Hierarchical clustering is performed on centered and scaled metric values using distance correlation and complete linkage. The cluster tree for metrics is cut at 6 groups. Bottom: An enlarged version of dendrogram corresponding to subjects with labels corresponding to diagnosis (diabetic or pre-diabetic).

Figure 6: Illustration of a graphical user interface for episode calculation functionality available via the accompanying Shiny App. The additional tab for episode calculation complements information presented in the standard AGP report.