quicR: An R Package for Real-Time Quaking Induced Conversion (RT-QuIC) Assays

Gage Rowden Peter Larsen

Abstract

Real-time quaking induced conversion (RT-QuIC) has quickly become an emerging diagnostic tool for protein misfolding disorders such as Creutzfeldt-Jakob disease and Parkinson's disease. Given that the technology is still relatively new, academic and industry standards have yet to be established. 'quicR' was developed to fill this lack of standardization by providing functions for data curation, analysis, and vizualiztion.

Introduction

Real-time quaking induced conversion is a cutting-edge diagnostic assay that has garnered significant attention for its ability to detect misfolded protein aggregates. The assay works by converting a recombinant protein substrate into an amyloid aggregate in the presence of a misfolded seed (Atarashi et al. 2011). The assay's sensitivity and specificity make RT-QuIC a promising tool for diagnosing diseases such as prion disorders and other protein misfolding pathologies. However, the relatively recent development and novelty of the assay have left a gap in widely accepted academic and industry standards for data analysis and interpretation.

To address this gap, we introduce quicR, an open-source R library dedicated to the manipulation, analysis, and visualization of RT-QuIC data. By consolidating key metrics and providing robust analytical tools, quicR aims to standardize the analysis pipeline and foster reproducibility within the field. quicR is designed with both researchers and diagnosticians in mind, providing a user-friendly interface that integrates seamlessly with existing R workflows.

While universal diagnostic criteria for RT-QuIC have yet to be established, certain analytical metrics have emerged as valuable tools for interpreting assay results. These include:

- Time-to-threshold (TtT): The time required for the fluorescence signal to exceed a predefined threshold, indicating amyloid formation (Orrú et al. 2015).
- Rate of amyloid formation (RAF): A measure of the kinetics of aggregate growth, which provides insight into relative misfolded seed quantity (Gallups and Harms 2022).
- Maxpoint ratio (MPR): A ratio-based metric for comparing peak normalized fluorescence intensities (Rowden et al. 2023).

- Maximum slope (MS): The steepest rate of fluorescence increase, reflecting the most rapid phase of aggregation (Henderson et al. 2015).

Together, these metrics enable researchers to characterize the kinetics of RT-QuIC reactions comprehensively, enhancing the rigor and reliability of diagnostic decisions.

In addition to analytical tools, quicR provides flexible and customizable visualization capabilities. Leveraging the powerful ggplot2 library (Wickham 2016), quicR enables users to generate high-quality, publication-ready figures. These visualizations can be further customized using the intuitive + syntax of ggplot2, allowing for tailored presentations of RT-QuIC data.

By combining standardized metrics, advanced visualization tools, and a commitment to open science, quicR serves as a foundational resource for the growing RT-QuIC community. Its goal is to empower researchers to analyze and present their data with clarity, consistency, and confidence.

Methods

Dependencies

This package requires the following dependencies: dplyr, ggplot2, janitor, openxlsx, readxl, reshape2, slider, stringr, and tidyr. Because the MARS software (BMG Labtech, Ortenberg, Germany) exports data as an Excel workbook, the packages, openxlsx (Schauberger and Walker 2024) and readxl (Wickham and Bryan 2023), were fundamental to performing downstream handling. The tidyverse packages (dplyr, ggplot2, stringr, and tidyr), were vital for writing easy-to-read code and for data visualization (Wickham et al. 2019). The janitor package (Firke 2024) has data cleaning functions which were useful when importing data from Excel. The slider package provides tools which apply some function to a moving window which was crucial for determining the approximate derivative of raw data (Vaughan 2024).

Input Data Formatting

MARS exports data into Excel workbooks. For many of the quicR functions to work together, the workbooks must be formatted correctly. In MARS, select “Excel Report”, and a pop-up window will appear (see Figure 1). The following options must be checked for proper output: “Microplate View”, “Table view”, “Transpose table”, and “Add test run information”. After clicking “Export report to Excel”, an Excel workbook is created such as in Figure 2.

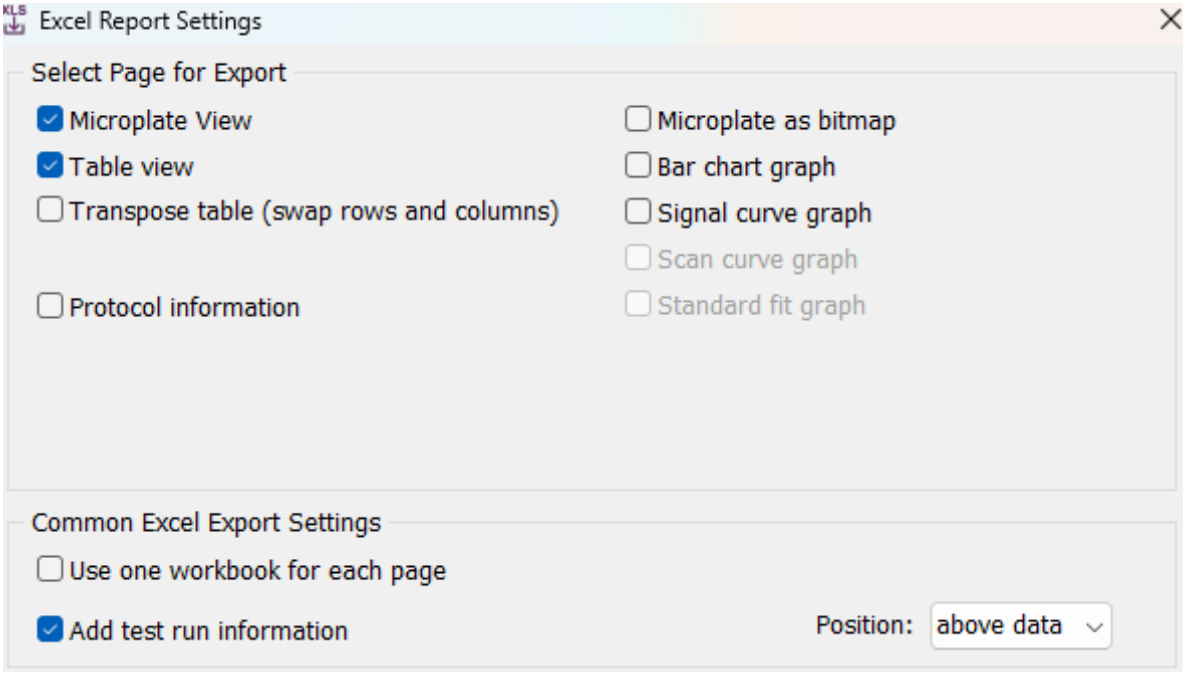


Figure 1: Excel export settings in MARS

User: USER
Path: C:\Program Files (x86)\BMG\Omega\User\Data
Test ID: 54
Test Name: RT-QuIC Plate Mode
Date: 4/24/2024
Time: 2:41:47 PM
ID1: 20240424_r4_PRC_Oral_Swabs
Fluorescence (FI)

1. Sample IDs

	1	2	3	4	5	6	7	8	9	10	11	12
A	P	S01	S02	S03	S04	S05	S06	S07	S08	S09	S10	S11
B	P	S01	S02	S03	S04	S05	S06	S07	S08	S09	S10	S11
C	P	S01	S02	S03	S04	S05	S06	S07	S08	S09	S10	S11
D	P	S01	S02	S03	S04	S05	S06	S07	S08	S09	S10	S11
E	N	S01	S02	S03	S04	S05	S06	S07	S08	S09	S10	S11
F	N	S01	S02	S03	S04	S05	S06	S07	S08	S09	S10	S11
G	N	S01	S02	S03	S04	S05	S06	S07	S08	S09	S10	S11
H	N	S01	S02	S03	S04	S05	S06	S07	S08	S09	S10	S11

(a) Microplate view Excel spreadsheet. This is the first spreadsheet in the workbook and contains the plate layouts of any data the user exported from MARS. Typically, there is a header containing metadata followed by a number of named matrices.

Well		A01	A02	A03	A04	A05	A06	A07
Content	Time [h]	Sample X1	Sample X2	Sample X3	Sample X4	Sample X5	Sample X6	Sample X7
Sample IDs		P	S01	S02	S03	S04	S05	S06
Raw Data (448-10/482-10)	0	813	664	670	643	685	396	677
Raw Data (448-10/482-10)	0.75	539	477	470	454	469	372	480
Raw Data (448-10/482-10)	1.5	564	476	484	470	486	375	481
Raw Data (448-10/482-10)	2.25	600	480	481	473	488	378	494
Raw Data (448-10/482-10)	3	622	476	478	474	487	368	486
Raw Data (448-10/482-10)	3.75	618	476	488	470	502	376	494
Raw Data (448-10/482-10)	4.5	623	482	486	479	484	368	499
Raw Data (448-10/482-10)	5.25	629	476	494	478	497	379	501
Raw Data (448-10/482-10)	6	618	483	490	477	493	365	506
Raw Data (448-10/482-10)	6.75	626	485	495	490	502	368	507
Raw Data (448-10/482-10)	7.5	624	482	495	479	502	368	508
Raw Data (448-10/482-10)	8.25	629	482	500	487	506	356	520
Raw Data (448-10/482-10)	9	626	487	500	483	511	364	523
Raw Data (448-10/482-10)	9.75	616	487	503	490	516	368	527
Raw Data (448-10/482-10)	10.5	613	481	490	490	520	366	519
Raw Data (448-10/482-10)	11.25	625	497	503	506	525	365	521
Raw Data (448-10/482-10)	12	616	492	498	496	518	363	520

(b) Table view Excel spreadsheet. This is the second spreadsheet in the workboook and contains columns of each well and time points. Each cell is a raw fluorescent value. This spreadsheet will also contain a header, but was excluded from this figure.

Figure 2: Excel spreadsheet exported from MARS.

Key Metrics and Calculations

quicR has functions for calculating TtT, MPR, and MS. A graphical representation of these can be found in Figure 3. There is no dedicated function for RAF since it can simply be expressed as the inverse of TtT, and can therefore be calculated separately as in Table 9.

TtT is calculated by iterating through each sample until a value is greater than the supplied threshold. It then determines the intersection between the previous and current read times and the threshold. If no value was found larger than the threshold, the total reaction run-time is returned.

MPR is defined as the maximum fluorescence divided by the background fluorescence. Thus, in order to calculate, the raw data must first be normalized against the background. This is done by choosing a cycle for background determination, and then dividing each read by that value. Once this is done, the MPR is taken as the max value.

Finally, MS is determined by approximating the maximum of the derivative of the raw data and is typically reported in units of ΔRFU/h (i.e. the change in relative fluorescent units per hour). Originally, this was accomplished by applying a linear regression to a sliding window; however, this proved to be very computationally expensive. Now, the slopes are calculated using differences between two data points within the range of the sliding window. While this slightly reduces the accuracy of the approximation, it was decided that the improvement in computation time was worth the loss in accuracy.

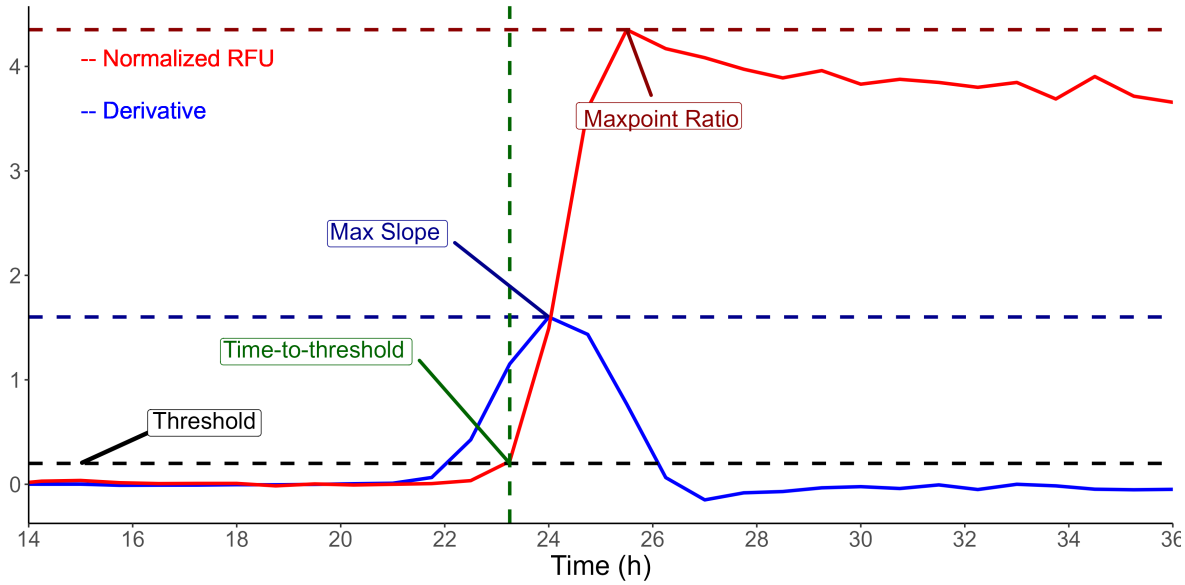


Figure 3: Example graphs highlighting the different calculated metrics. The red curve represents a raw data curve that has been normalized against background. MPR is calculated as the maximum fluorescent value achieved in the normalized raw data. TtT is determined as the time required to cross a given threshold (in this example, the threshold is 0.2). The blue curve represents the derivative of the raw data, and max slope is determined as the maximum of the derivative.

Development

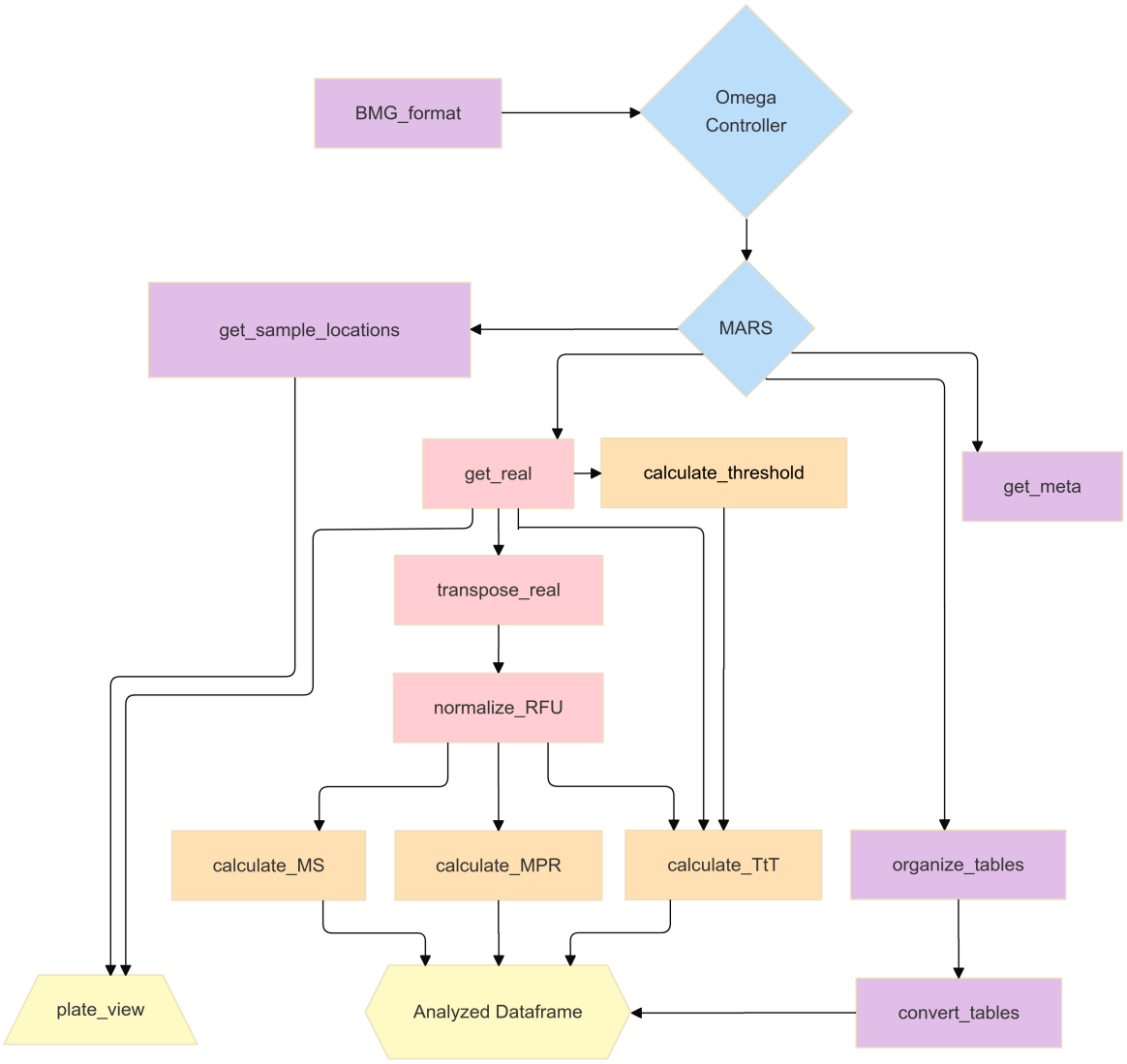


Figure 4: Workflow hierarchy of the quicR package. Blue nodes indicate steps where BMG software is needed. Purple nodes indicate functions dedicated to handling metadata. Red nodes are functions that acquire and manipulate raw data. Orange nodes are functions which calculate some metric. Finally, yellow nodes represent endpoints in the data analysis.

quicR was developed to address the need for efficient data conversion and analysis of RT-QuIC data. The functions were designed with usability and reproducibility in mind, ensuring compatibility between multiple labs. Currently, the package accepts data exported from the proprietary MARS software (BMG Labtech, Ortenberg, Germany) as an Excel workbook.

This project was devloped using Git version control. GitHub workflows were utilized to en-

sure that the package is able to be installed on current operating systems. Additionally, tests were performed on each function using the testthat package (Wickham 2011).

The functionality in this package revolves around data curation, metric calculations, and visualization.

Implementation

The implementation of the quicR package encompasses several streamlined processes designed to facilitate data input, cleaning, transformation, and analysis of real-time fluorescence data. This section provides a comprehensive guide to utilizing the package's key functionalities, detailing how to:

1. Format and input sample data into MARS.
2. Extract, clean, and organize metadata and raw fluorescence data.
3. Apply transformations and normalization to raw data for downstream analysis.
4. Calculate critical analytical metrics, such as time-to-threshold (TtT), rate of amyloid formation (RAF), maxpoint ratio (MPR), and maximum slope (MS).

These steps are designed to enhance reproducibility, minimize manual data handling, and enable seamless integration with the MARS software workflow. Through practical examples, this section illustrates how each function operates, along with expected input and output formats, ensuring clarity and ease of use for researchers.

Input of Sample IDs into MARS

The Omega control software allows input of a TXT file containing sample IDs, dilution factors, and their well locations. This file is uniquely formatted, and not easily reproduced manually. The function, "BMG_format", allows for input of a CSV file containing the plate layout (see Table 1 for proper formatting), and exports the formatted TXT file. The file can then be imported into the control software before running.

Formatted Plate Layout for MARS Input

The function, "BMG_format", includes the logical argument "write_file". If TRUE, it will create a TXT file. The path can be given to the "save_path" argument, and the file name can be supplied to the "save_name" argument. The text file will be formatted as follows, and can be imported into MARS.

```
BMG_format(sample_file, write_file = TRUE)
```

```
A1  P       P
B1  P       P
C1  P       P
```

Table 1: Example CSV file plate layout for input into the "BMG_format" function. The top left corner should be cell "A1" in the CSV file. The top numbered row and the left-most lettered column should never be altered.

	1	2	3	4	5	6	7	8	9	10	11	12
A	P	S01	S02	S03	S04	S05	S06	S07	S08	S09	S10	S11
B	P	S01	S02	S03	S04	S05	S06	S07	S08	S09	S10	S11
C	P	S01	S02	S03	S04	S05	S06	S07	S08	S09	S10	S11
D	P	S01	S02	S03	S04	S05	S06	S07	S08	S09	S10	S11
E	N	S01	S02	S03	S04	S05	S06	S07	S08	S09	S10	S11
F	N	S01	S02	S03	S04	S05	S06	S07	S08	S09	S10	S11
G	N	S01	S02	S03	S04	S05	S06	S07	S08	S09	S10	S11
H	N	S01	S02	S03	S04	S05	S06	S07	S08	S09	S10	S11

```
D1  P       P
E1  N       N
F1  N       N
G1  N       N
H1  N       N
A2  X1      S01
B2  X1      S01
C2  X1      S01
D2  X1      S01
E2  X1      S01
F2  X1      S01
G2  X1      S01
H2  X1      S01
```

Data Cleaning and Transformation

The MARS software (BMG Labtech, Ortenberg, Germany) exports real-time data as an Excel workbook. Typically, the first sheet in the workbook will include microplate views of both raw data and metadata; however, the metadata on this page is what will be most useful. Those tables are the "Sample IDs" and the "Dilutions" tables (if dilutions were included in the export). For much of the downstream analysis, it is crucial the the "Sample IDs" table was exported. If there is no table, the user can simply add it manually (see Figure 2a for proper formatting).

Retrieving Metadata

The metadata is defined as either sample-dependent or -independent. Sample-dependent metadata includes information such as sample IDs and dilution factors, whereas sample-independent metadata includes the date, time, reaction ID, etc.

The dependent metadata can be retrieved using the "organize_tables" and "convert_tables" functions. The former returns a list of tables, and the latter converts each table into a column in a single data frame. See Table 2 and Table 3 for example outputs of these functions.

```
organize_tables(file)
```

Table 2: Sample IDs and Dilution Factors

	1	2	3	4	5	6	7	8	9	10	11	12
A	P	S01	S02	S03	S04	S05	S06	S07	S08	S09	S10	S11
B	P	S01	S02	S03	S04	S05	S06	S07	S08	S09	S10	S11
C	P	S01	S02	S03	S04	S05	S06	S07	S08	S09	S10	S11
D	P	S01	S02	S03	S04	S05	S06	S07	S08	S09	S10	S11
E	N	S01	S02	S03	S04	S05	S06	S07	S08	S09	S10	S11
F	N	S01	S02	S03	S04	S05	S06	S07	S08	S09	S10	S11
G	N	S01	S02	S03	S04	S05	S06	S07	S08	S09	S10	S11
H	N	S01	S02	S03	S04	S05	S06	S07	S08	S09	S10	S11

	1	2	3	4	5	6	7	8	9	10	11	12
A	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000
B	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000
C	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000
D	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000
E	1000	10000	10000	10000	10000	10000	10000	10000	10000	10000	10000	10000
F	1000	10000	10000	10000	10000	10000	10000	10000	10000	10000	10000	10000
G	1000	10000	10000	10000	10000	10000	10000	10000	10000	10000	10000	10000
H	1000	10000	10000	10000	10000	10000	10000	10000	10000	10000	10000	10000

```
organize_tables(file) %>% convert_tables()
```

Table 3: The tables extracted from the "organize_tables" function are converted into columns in a dataframe. This format is much more compatible with downstream analysis.

Sample IDs	Dilutions
P	1000
S01	1000
S02	1000
S03	1000
S04	1000

Table 3: The tables extracted from the "organize_tables" function are converted into columns in a dataframe. This format is much more compatible with downstream analysis.

Sample IDs	Dilutions
S05	1000
S06	1000
S07	1000
S08	1000
S09	1000
S10	1000
S11	1000

Sample Locations

Samples locations can be extracted based on their well ID. The "get_sample_locations" function accepts additional arguments if dilution factors were exported from MARS. The "dilution_fun" argument will supply a function for transforming the dilution factors (e.g. if the user would want to perform a log transformation). This output of this function is critical as an argument in "plate_view" function which is further explained in the visualization section.

```
get_sample_locations(
  file,
  dilution_bool = TRUE,
  dilution_fun = function(x)
    -log10(x),
  sep = " "
)
```

V1	IDs
A01	P -3
A02	S01 -3
A03	S02 -3
A04	S03 -3
A05	S04 -3
A06	S05 -3
A07	S06 -3
A08	S07 -3
A09	S08 -3
A10	S09 -3
A11	S10 -3
A12	S11 -3

Sample-Independent Metadata

The independent metadata can be retrieved using the "get_meta" function. This data is included in the header of the excel workbook.

```
get_meta(file)
```

Meta_ID	Meta_info
User	USER
Path	C:/Program Files (x86)/BMG/Omega/User/Data
Test ID	54
Test Name	RT-QuIC Plate Mode
Date	4/24/2024
Time	2:41:47 PM
ID1	20240424_r4_PRC_Oral_Swabs
Fluorescence (FI)	NA

Retrieving and Manipulating Raw Data

The raw, real-time data is typically found on the second sheet of the Excel workbook. There are three functions dedicated to the retrieval and cleaning of raw data. 1. get_real: Retrieves the raw data from the Excel file, and outputs it as a dataframe. 2. transpose_real: Swaps the rows and columns which makes some downstream analyses easier. 3. normalize_RFU: normalizes the raw data by dividing each read by background fluorescence at a given cycle.

Retrieve Raw Data

Raw data can be retrieved using the "get_real" function. The logical argument, "ordered", indicates whether the user would prefer the columns to be ordered by well or by sample ID. By default, it is FALSE which will order the data by well. This should almost always be the case for easier integration with other downstream functions. Additionally, since there can be more than one instance of real-time data (depending on if the user added some calculations in MARS), "get_real" returns a list of dataframes. Therefore, the output should be indexed to access the data frame of interest.

```
get_real(file)[[1]]
```

Time	sample_x1	sample_x2	sample_x3	sample_x4	sample_x5	sample_x6	sample_x7
0	813	664	670	643	685	396	677
0.75	539	477	470	454	469	372	480
1.5	564	476	484	470	486	375	481
2.25	600	480	481	473	488	378	494
3	622	476	478	474	487	368	486

Time	sample_x1	sample_x2	sample_x3	sample_x4	sample_x5	sample_x6	sample_x7
3.75	618	476	488	470	502	376	494
4.5	623	482	486	479	484	368	499
5.25	629	476	494	478	497	379	501
6	618	483	490	477	493	365	506
6.75	626	485	495	490	502	368	507
7.5	624	482	495	479	502	368	508

Transpose Raw Data

This data is structured such that each sample is its own column (variable) and each row (observation) is a time point. While this format is technically correct, a transposed format is more ideal for some downstream manipulation. This operation is performed using the function, "transpose_real". After transposition, each time point is an individual column (variable), and each sample is an individual row (observation).

```
get_real(file)[[1]] %>% transpose_real()
```

Sample IDs	0	0.75	1.5	2.25	3	3.75	4.5	5.25	6	6.75	7.5
sample_x1	813	539	564	600	622	618	623	629	618	626	624
sample_x2	664	477	476	480	476	476	482	476	483	485	482
sample_x3	670	470	484	481	478	488	486	494	490	495	495
sample_x4	643	454	470	473	474	470	479	478	477	490	479
sample_x5	685	469	486	488	487	502	484	497	493	502	502
sample_x6	396	372	375	378	368	376	368	379	365	368	368
sample_x7	677	480	481	494	486	494	499	501	506	507	508

Normalize Raw Data

The function "normalize_RFU" will convert the raw data into a background normalized data set. The function includes two additional arguments, "bg_cycle" (the cycle which will be used as the background fluorescence value) and "transposed" (if FALSE, will make a call to the "transpose_real" function). Note that the fourth time point is all "1's" since this was designated the background cycle.

```
get_real(file)[[1]] %>% normalize_RFU(transposed = FALSE)
```

Sample IDs	0	0.75	1.5	2.25	3	3.75	4.5	5.25	6	6.75	7.5
sample_x1	1.35	0.90	0.94	1	1.04	1.03	1.04	1.05	1.03	1.04	1.04

Sample IDs	0	0.75	1.5	2.25	3	3.75	4.5	5.25	6	6.75	7.5
sample_x2	1.38	0.99	0.99	1	0.99	0.99	1.00	0.99	1.01	1.01	1.00
sample_x3	1.39	0.98	1.01	1	0.99	1.01	1.01	1.03	1.02	1.03	1.03
sample_x4	1.36	0.96	0.99	1	1.00	0.99	1.01	1.01	1.01	1.04	1.01
sample_x5	1.40	0.96	1.00	1	1.00	1.03	0.99	1.02	1.01	1.03	1.03
sample_x6	1.05	0.98	0.99	1	0.97	0.99	0.97	1.00	0.97	0.97	0.97
sample_x7	1.37	0.97	0.97	1	0.98	1.00	1.01	1.01	1.02	1.03	1.03

Calculations

There are three analytical metrics with dedicated functions: time-to-threshold (TtT), max-point ratio (MPR), and maximum slope (MS). The rate of amyloid formation does not have a designated function since it is simply the inverse of the time-to-threshold. Each function below accepts input from the "transpose_real" or the "normalize_RFU" functions. See Table 9 for an example of the output of these functions.

Thresholds

Many publications have different methods of determining thresholds. By and large, the most popular method is to take the average background fluorescence of the entire plate and add some multiple of standard deviations. The quicR package provides the "calculate_threshold" function for this purpose. A value can be provided as the optional argument, "multiplier", which will be applied to the standard deviation.

```
get_real(file)[[1]] %>%
  calculate_threshold(method = "stdev", multiplier = 10)
```

```
[1] 919.7532
```

Time-to-threshold & Rate of Amyloid Formation

TtT is calculated using the "calculate_TtT" function. The function must be supplied a threshold; default value is 2 (i.e. twice the background fluorescence if the data is normalized). A starting column should also be given as an integer; default value is 3. This is essentially asking how many columns of metadata are included before the fluorescence reads begin.

TtT is calculated by iterating through each row and checking if a value is greater than the threshold. If the value is greater, the slope of the previous time-point to the current time-point is calculated, and the time intersection of the current read is returned.

Maxpoint Ratio

MPR is calculated by the "calculate_MPR" function. Data must be normalized in order to derive this metric. In a normalized data set, the MPR is simply the maximum value achieved during the run. Raw data can be passed to this function, but the argument, "data_is_norm", must be set to TRUE. This will pass the raw data to "normalize_RFU" before calculating the MPR values.

Maximum Slope

MS is calculated by the "calculate_MS" function. The function iterates through each row using a rolling window which can be adjusted (default value is 3). Given the window size, the slope is calculated based on change in fluorescence divided by the range of the window. The MS is simply the largest slope value recorded. The units are typically reported as ΔRFU/h.

```
df_norm <- get_real(file) %>% normalize_RFU()

data.frame("Sample IDs" = tabs$`Sample IDs`) %>%
  mutate(
    Dilutions = -log10(tabs$dilutions),
    MPR = calculate_MPR(df_norm, start_col = 3, data_is_norm = TRUE),
    MS  = calculate_MS(df_norm, data_is_norm = TRUE),
    TtT = calculate_TtT(df_norm, threshold = 2, start_col = 3),
    RAF = 1 / TtT
  )
```

Table 9: Calculated metrics

Sample IDs	Dilutions	MPR	MS	TtT	RAF
P	-3	5.35	1.836	23.71	0.0422
S01	-3	1.05	0.174	48.00	0.0208
S02	-3	1.06	0.185	48.00	0.0208
S03	-3	1.07	0.178	48.00	0.0208
S04	-3	1.08	0.197	48.00	0.0208
S05	-3	1.99	0.450	48.00	0.0208
S06	-3	1.07	0.177	48.00	0.0208
S07	-3	1.07	0.175	48.00	0.0208
S08	-3	1.10	0.179	48.00	0.0208
S09	-3	1.06	0.169	48.00	0.0208
S10	-3	1.08	0.182	48.00	0.0208
S11	-3	1.12	0.155	48.00	0.0208

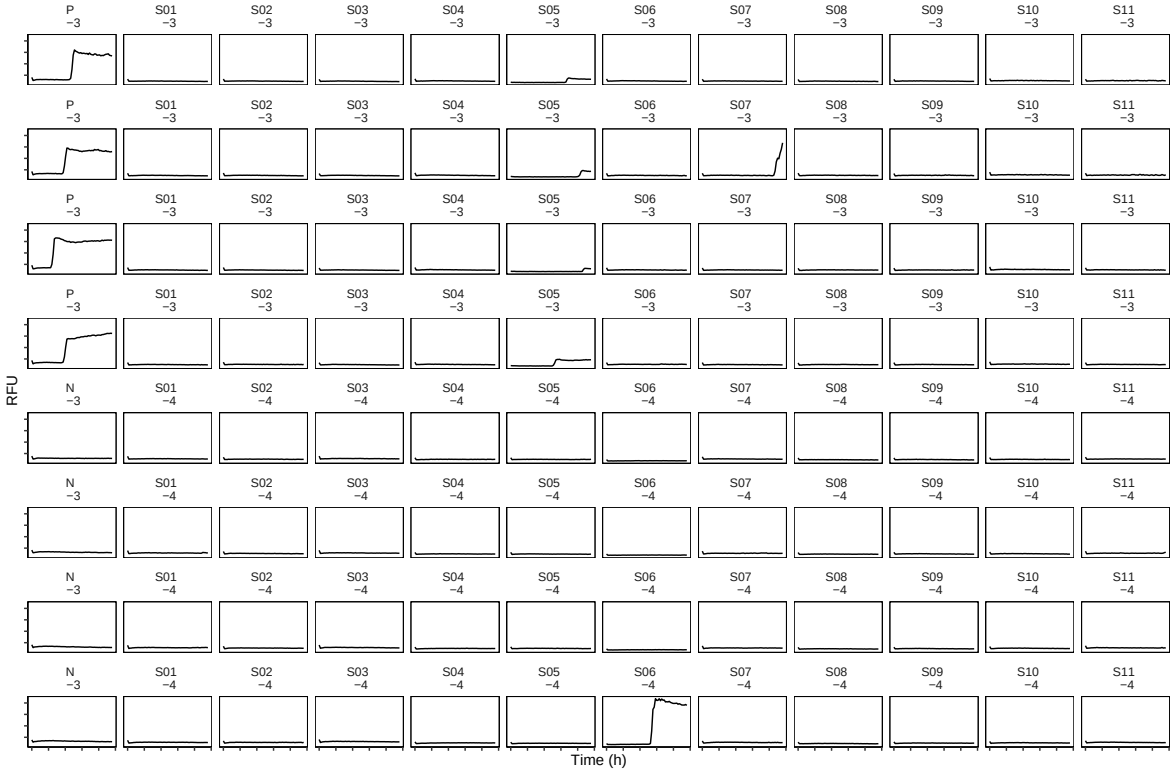
Visualization

Plate View

The “plate_view” function requires *un-transposed* data and sample locations as arguments. It also has an argument for plate type which will either be 96 or 384.

```
sample_locations <- get_sample_locations(
  file,
  dilution_bool = TRUE,
  dilution_fun = function(x) -log10(x)
)

plate_view(df_, sample_locations)
```

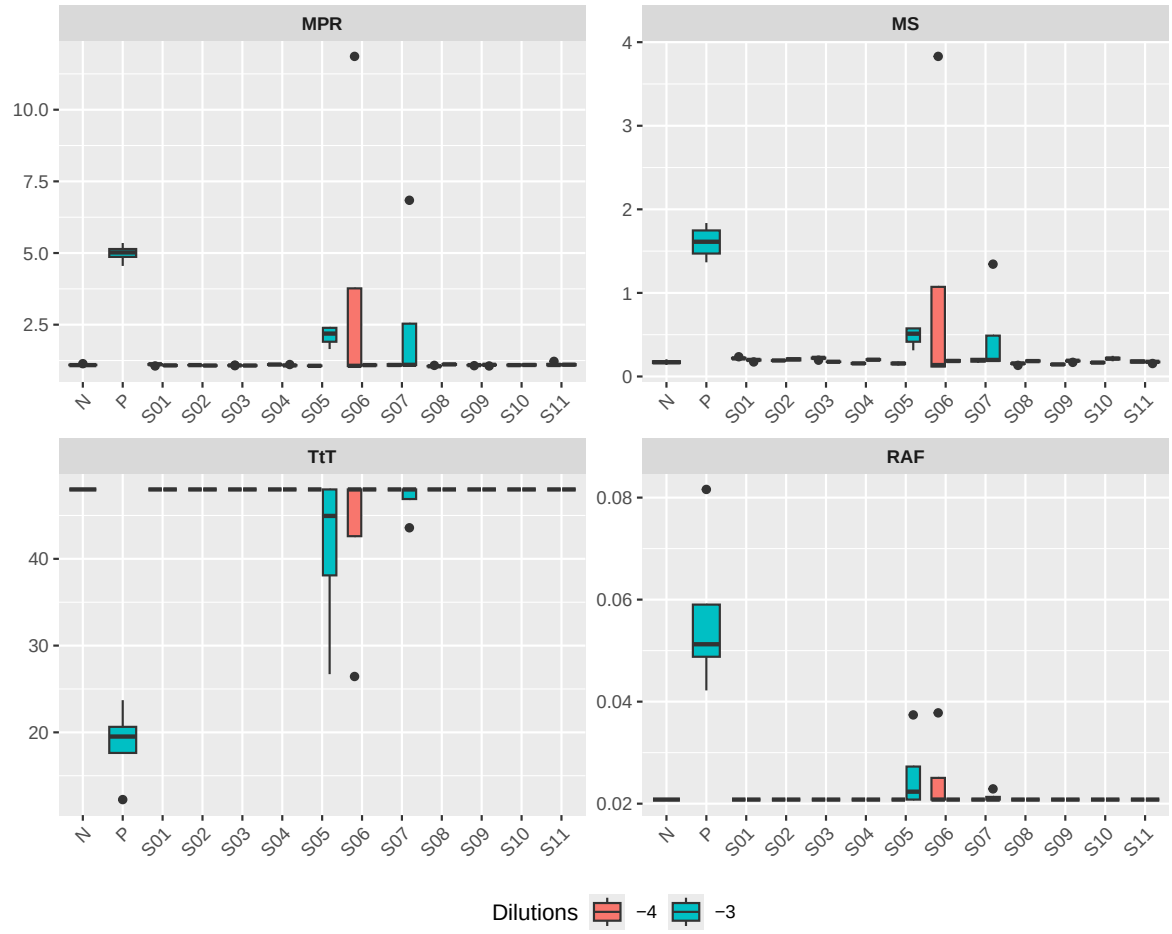


Summary Plots

```
df_analyzed %>%
  melt(id.vars = c("Sample IDs", "Dilutions")) %>%
  mutate_at("Dilutions", as.factor) %>%

  ggplot(aes(`Sample IDs`, value, fill = Dilutions)) +
    geom_boxplot() +
```

```
  facet_wrap(~variable, scales = "free") +
  theme(
    legend.position = "bottom",
    strip.text = element_text(face = "bold"),
    axis.text.x = element_text(angle = 45, hjust = 1, vjust = 1),
    axis.title = element_blank()
  )
```



Usage

Installation

```
# Latest CRAN release
install.packages("quicR")

# Development version
```

```
devtools::install_github("gage1145/quicR")
```

Validation & Performance

Discussion

Conclusion

quicR provides improved and standardized methods for analyzing RT-QuIC data.

Acknowledgments

References

Atarashi, Ryuichiro, Kazunori Sano, Katsuya Satoh, and Noriyuki Nishida. 2011. "Real-Time Quaking-Induced Conversion: A Highly Sensitive Assay for Prion Detection." *Prion*. Taylor & Francis. https://doi.org/10.4161/pri.5.3.16893.

Firke, Sam. 2024. *Janitor: Simple Tools for Examining and Cleaning Dirty Data*. https://CRAN.R-project.org/package=janitor.

Gallups, Nicole J., and Ashley S. Harms. 2022. "'Seeding' the Idea of Early Diagnostics in Synucleinopathies." *Brain* 145 (April): 418–19. https://doi.org/10.1093/BRAIN/AWAC062.

Henderson, Davin M., Kristen A. Davenport, Nicholas J. Haley, Nathaniel D. Denkers, Candace K. Mathiason, and Edward A. Hoover. 2015. "Quantitative Assessment of Prion Infectivity in Tissues and Body Fluids by Real-Time Quaking-Induced Conversion." *Journal of General Virology* 96 (January): 210–19. https://doi.org/10.1099/vir.0.069906-0.

Orrú, Christina D., Bradley R. Groveman, Andrew G. Hughson, Gianluigi Zanusso, Michael B. Coulthart, and Byron Caughey. 2015. "Rapid and Sensitive RT-QuIC Detection of Human Creutzfeldt-Jakob Disease Using Cerebrospinal Fluid." *mBio* 6 (January). https://doi.org/10.1128/mBio.02451-14.

Rowden, Gage R., Catalina Picasso-Risso, Manci Li, Marc D. Schwabenlander, Tiffany M. Wolf, and Peter A. Larsen. 2023. "Standardization of Data Analysis for RT-QuIC-Based Detection of Chronic Wasting Disease." *Pathogens* 12 (February): 309. https://doi.org/10.3390/PATHOGENS12020309/S1.

Schauberger, Philipp, and Alexander Walker. 2024. *Openxlsx: Read, Write and Edit Xlsx Files*. https://CRAN.R-project.org/package=openxlsx.

Vaughan, Davis. 2024. *Slider: Sliding Window Functions*. https://CRAN.R-project.org/package=slider.

Wickham, Hadley. 2011. “Testthat: Get Started with Testing.” *The R Journal* 3: 5–10. https://journal.r-project.org/archive/2011-1/RJournal_2011-1_Wickham.pdf.

———. 2016. *Ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York. https://ggplot2.tidyverse.org.

Wickham, Hadley, Mara Averick, Jennifer Bryan, Winston Chang, Lucy D’Agostino McGowan, Romain François, Garrett Grolemund, et al. 2019. “Welcome to the tidyverse.” *Journal of Open Source Software* 4 (43): 1686. https://doi.org/10.21105/joss.01686.

Wickham, Hadley, and Jennifer Bryan. 2023. *Readxl: Read Excel Files*. https://CRAN.R-project.org/package=readxl.