quicR: An R Library for Streamlined Data Handling of Real-Time Quaking Induced Conversion Assays

Gage Rowden[1,2,3], Peter Larsen[1,2,3]

[1]Department of Veterinary and Biomedical Sciences, University of Minnesota, St. Paul, MN 55108, USA.
[2]Minnesota Center for Prion Research and Outreach, University of Minnesota, St. Paul, MN 55108, USA.
[3]Priogen Corp., St. Paul, MN 55114, USA.

Abstract

Real-time quaking induced conversion (RT-QuIC) has quickly become a valuable diagnostic tool for protein misfolding disorders such as Creutzfeldt-Jakob disease and Parkinson's disease. Given that the technology is relatively new, academic and industry standards for quality filtering data and high throughput analysis of results have yet to be fully established. The open source R library, quicR, was developed to provide a standradized approach to RT-QuIC data analysis. quicR provides functions, which can be easily integrated into existing R workflows, for data curation, analysis, and vizualization.

Introduction

Real-time quaking induced conversion (RT-QuIC) is a cutting-edge diagnostic assay that has garnered significant attention for its ultra-sensitive detection of misfolded protein aggregates (Wilham et al. 2010; Atarashi et al. 2011). The assay works by converting a recombinant protein substrate into an amyloid aggregate in the presence of a misfolded seed (Wilham et al. 2010; Orrú et al. 2012, 2015; Orrù et al. 2017; Bongianni et al. 2019; Dassanayake et al. 2016; Hwang et al. 2018; Groveman et al. 2018; Metrick et al. 2020). The assay's sensitivity and specificity make RT-QuIC a promising tool for diagnosing diseases such as prion disorders and other protein misfolding pathologies (Fiorini et al. 2020; Franceschini et al. 2017; Picasso-Risso et al. 2022; Holz et al. 2021). 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 (Rowden et al. 2023).

To address this gap, we introduce quicR, an open-source library, developed in R (R Core Team 2024), dedicated to the cleaning, 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 of quaking induced assays including related assays such as Nano-QuIC (Christenson et al. 2023) and Micro-QuIC (Lee et al. 2024). 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 and kinetics. These include:

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

Methods

Dependencies

This package requires the following dependencies: dplyr, ggplot2, stringr, tidyr, janitor, openxlsx, readxl, reshape2, and slider. The packages, openxlsx (Schauberger and Walker 2024) and readxl (Wickham and Bryan 2023), were fundamental to performing initial data handling of raw data Excel files. 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 (Vaughan 2024) provides tools which apply some function to a moving window which was crucial for determining the approximate derivative of raw data.

Input Formatting

The FLUOstar® Omega series microplate readers (BMG Labtech, Ortenberg, Germany) are by far the most common readers used for RT-QuIC. As such, their analysis software, MARS, is integral to data input for this package. MARS exports data into Excel workbooks, and 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 (Figure 1) will appear. The following options must be checked for proper output: "Microplate View", "Table view", "Transpose table", and "Add test run information".

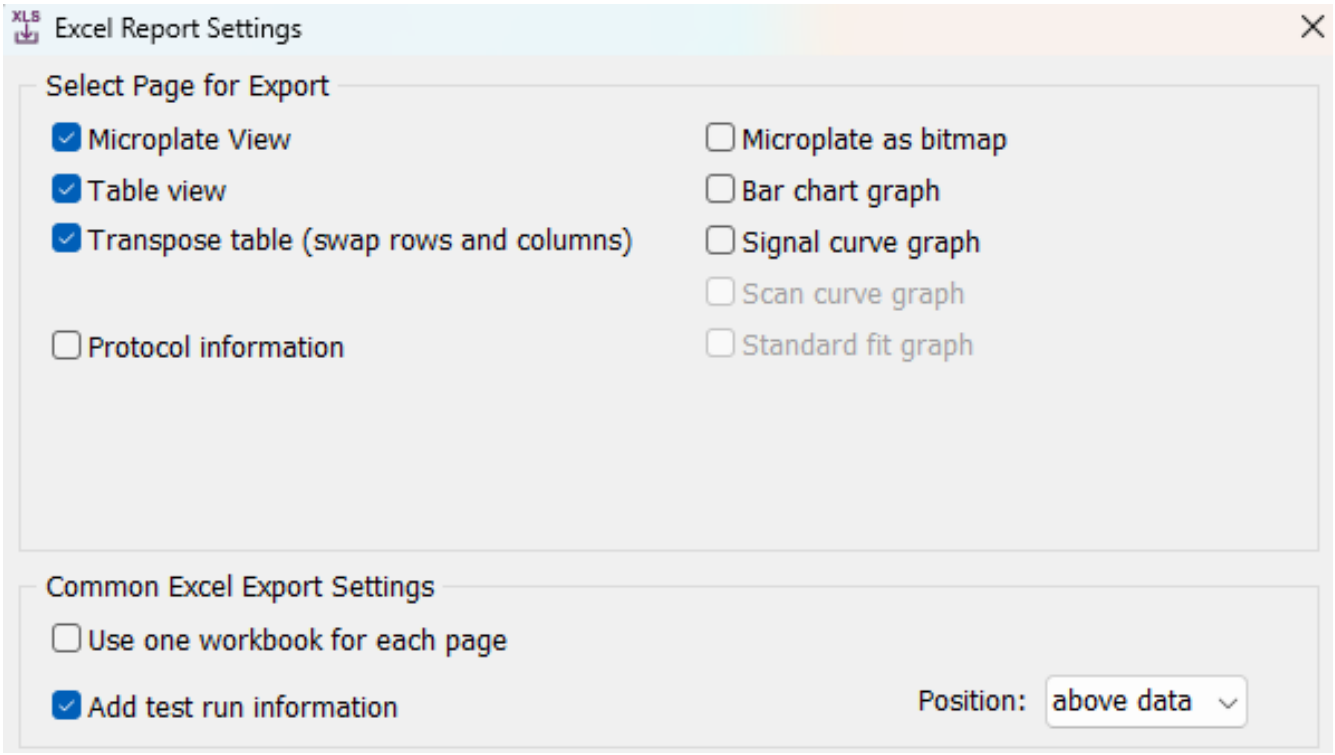


Figure 1: Excel export settings in MARS. Ensure that "Microplate View", "Table view", "Transpose table", and "Add test run information" are selected.

After clicking "Export report to Excel", an Excel workbook is created such as in Figure 2. This workbook will contain two sheets, the first with the microplate views and the second with the table view. The first sheet will have all of the relevant metadata while the second sheet will have all of the raw fluorescent data.

User: USER												
Path: C:\Program Files (x86)\BMG\Omega\User\Data												
Test ID: 183												
Test Name: RT-QuIC Plate Mode												
Date: 7/22/2024												
Time: 12:31:15 PM												
ID1: 20240722_r1_GR												
Fluorescence (FI)												
	1. Sample IDs											
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
A	P	Sample 01	Sample 02	Sample 03	Sample 04	Sample 05	Sample 06	Sample 07	Sample 08	Sample 09	Sample 10	Sample 11
B	P	Sample 01	Sample 02	Sample 03	Sample 04	Sample 05	Sample 06	Sample 07	Sample 08	Sample 09	Sample 10	Sample 11
C	P	Sample 01	Sample 02	Sample 03	Sample 04	Sample 05	Sample 06	Sample 07	Sample 08	Sample 09	Sample 10	Sample 11
D	P	Sample 01	Sample 02	Sample 03	Sample 04	Sample 05	Sample 06	Sample 07	Sample 08	Sample 09	Sample 10	Sample 11
E	N	Sample 01	Sample 02	Sample 03	Sample 04	Sample 05	Sample 06	Sample 07	Sample 08	Sample 09	Sample 10	Sample 11
F	N	Sample 01	Sample 02	Sample 03	Sample 04	Sample 05	Sample 06	Sample 07	Sample 08	Sample 09	Sample 10	Sample 11
G	N	Sample 01	Sample 02	Sample 03	Sample 04	Sample 05	Sample 06	Sample 07	Sample 08	Sample 09	Sample 10	Sample 11
H	N	Sample 01	Sample 02	Sample 03	Sample 04	Sample 05	Sample 06	Sample 07	Sample 08	Sample 09	Sample 10	Sample 11

(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]	control P	Sample X1	Sample X3	Sample X5	Sample X7	Sample X9	Sample X11
Dilutions		1000	1000	1000	1000	1000	1000	1000
Sample IDs		P	Sample_1	Sample_2	Sample_3	Sample_4	Sample_5	Sample_6
Raw Data (448-10/482-10)	0	748	749	740	726	720	718	675
Raw Data (448-10/482-10)	0.75	503	501	498	508	501	509	458
Raw Data (448-10/482-10)	1.5	564	537	515	533	523	526	460
Raw Data (448-10/482-10)	2.25	601	555	531	557	521	524	452
Raw Data (448-10/482-10)	3	604	559	515	557	524	523	458
Raw Data (448-10/482-10)	3.75	626	565	525	562	516	524	452
Raw Data (448-10/482-10)	4.5	634	560	521	565	512	518	439
Raw Data (448-10/482-10)	5.25	635	551	515	571	507	518	448
Raw Data (448-10/482-10)	6	654	572	513	567	511	515	439
Raw Data (448-10/482-10)	6.75	652	568	511	572	501	519	432
Raw Data (448-10/482-10)	7.5	652	566	501	573	507	516	435
Raw Data (448-10/482-10)	8.25	648	568	505	568	498	505	426
Raw Data (448-10/482-10)	9	645	557	630	561	494	510	419
Raw Data (448-10/482-10)	9.75	652	563	2945	556	495	506	414
Raw Data (448-10/482-10)	10.5	658	556	6259	560	493	502	432
Raw Data (448-10/482-10)	11.25	648	546	6545	555	488	502	426
Raw Data (448-10/482-10)	12	643	552	6570	556	492	500	411

(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 workbook exported from MARS.

Key Metrics and Calculations

quicR has functions for calculating TtT, MPR, and MS (vizualized in Figure 3). There is no dedicated function for RAF since it can be expressed as the inverse of TtT, and can therefore be calculated separately as in the example in Table 10.

TtT is calculated by iterating through each sample until a value is greater than the user-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 (Rowden et al. 2023). Thus, in order to calculate, the raw data must first be normalized against the background. This is done by the user choosing a cycle for background determination, and then dividing each read by that value. The MPR is taken as the max value of the normalized data.

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, slopes are calculated using differences between two data points within the range of a sliding window. While this slightly reduces the accuracy of the approximation, the improvement in computation time exceeded the loss in resolution.

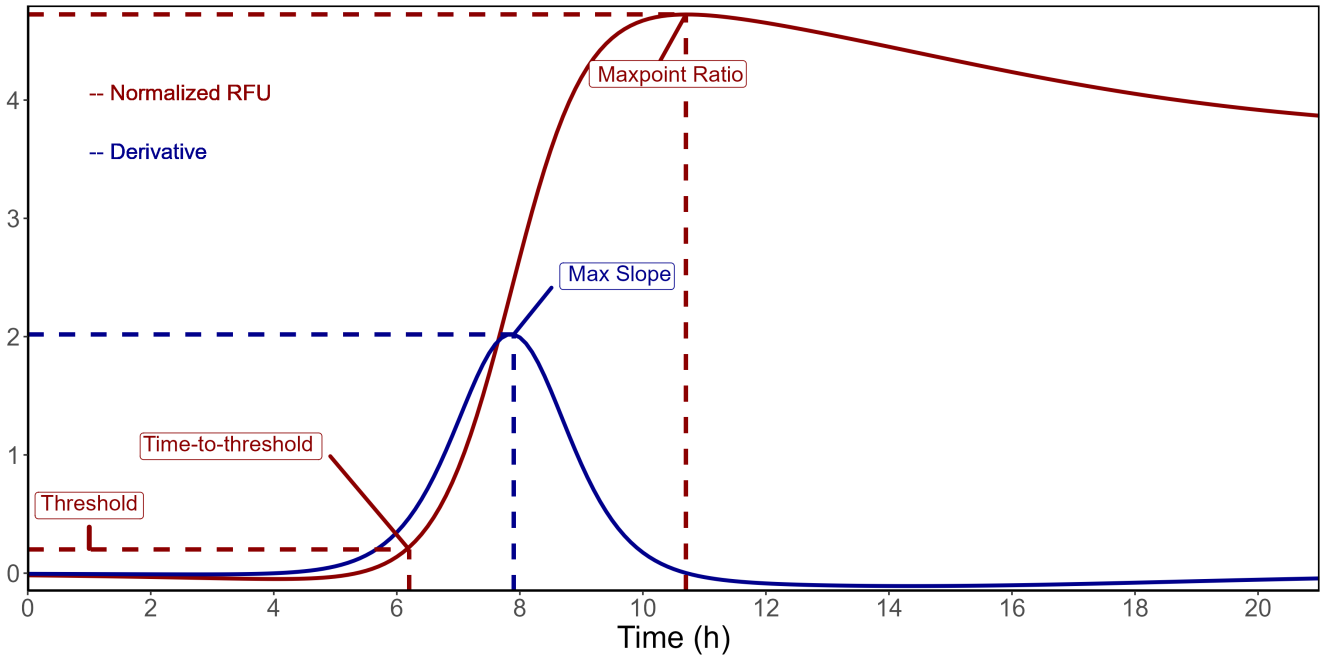


Figure 3: Example graph highlighting the calculated metrics described above. The red curve represents a raw data curve that has been normalized against background. The maxpoint ratio is calculated as the maximum fluorescent value achieved in the normalized raw data. Time-to-threshold is determined as the time required to cross a given threshold (in this example, the threshold is set at 0.2). The blue curve represents the derivative of the raw data, and max slope is determined as the maximum of the derivative.

Development

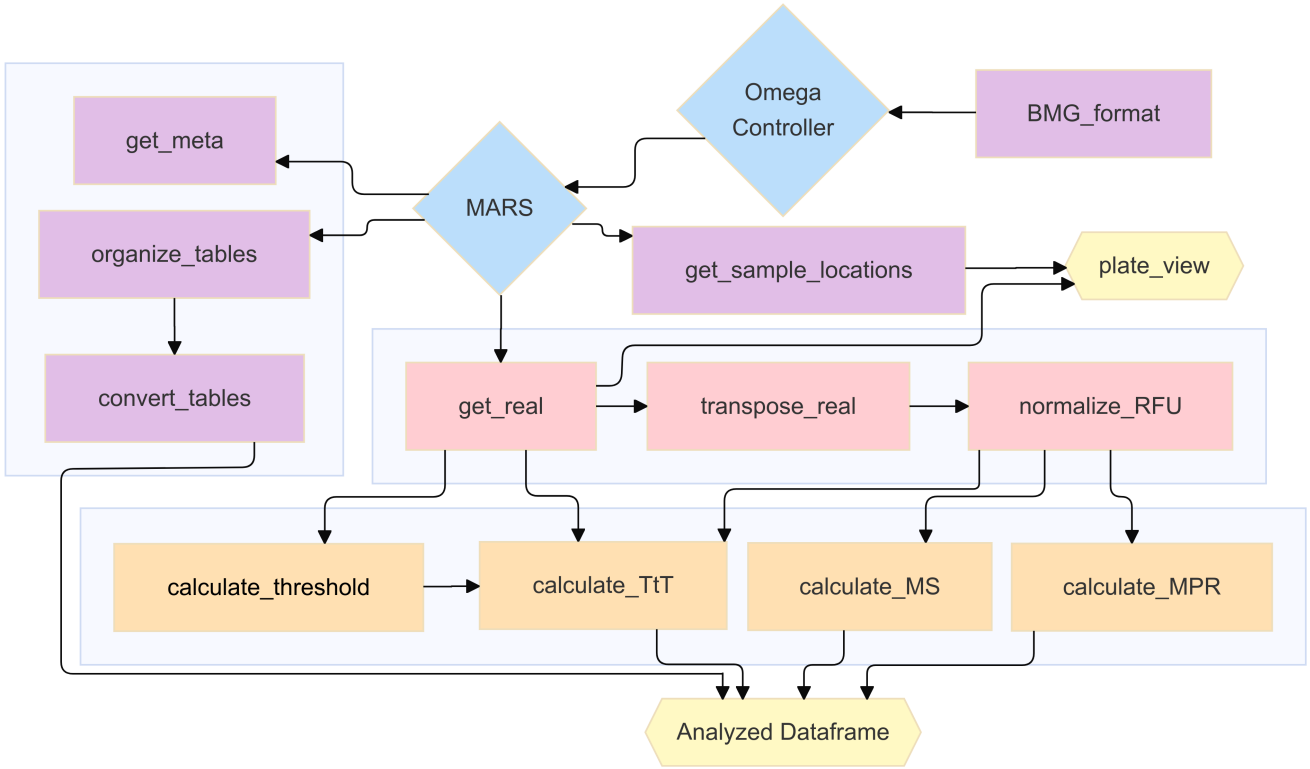


Figure 4: Workflow hierarchy of the quicR package. Blue nodes indicate steps where BMG software is needed. Purple nodes indicate functions dedicated to handling metadata. Red nodes are functions that acquire and manipulate raw data. Orange nodes are functions which calculate some metric. Finally, yellow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

Version Control and Collaboration

The project was managed using Git for version control, enabling efficient tracking of changes and fostering collaborative development. The source code is hosted on GitHub (https://github.com/gage1145/quicR), ensuring transparency and encouraging contributions from the research community. GitHub Actions were employed to automate testing and deployment workflows, verifying that the package remains compatible with current versions of R and its dependencies.

Testing and Validation

Robust testing protocols were implemented to guarantee the reliability of quicR's functionality. Each function was subjected to comprehensive unit testing using the testthat package (Wickham 2011). These tests ensure that key metrics, data manipulation routines, and visualization tools perform as expected, even as the package evolves. Each new build of quicR must pass these tests in order to be released.

Core Functionalities

The functionality of quicR is centered around three primary objectives:

1. Data Curation: Functions for importing, cleaning, and normalizing raw RT-QuIC data to streamline pre-processing.
2. Metric Calculations: Tools for deriving critical metrics, such as time-to-threshold (TtT), maxpoint ratio (MPR), and maximum slope (MS), to enable standardized data interpretation.
3. Visualization: High-quality plotting capabilities powered by the ggplot2 library, enabling users to create publication-ready figures with minimal effort.

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 Omega control software (BMG Labtech, Ortenberg, Germany).
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 Omega Control Software

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.

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

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(file, write_file = TRUE, save_path = "", save_name = "formatted.txt")
```

```
A1  P     P
B1  P     P
C1  P     P
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 is most useful for downstream processes. Those tables include the "Sample IDs" and "Dilutions" tables (if dilutions were included in the MARS export). For much of the downstream analysis, it is crucial that the "Sample IDs" table was exported from MARS. 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. Note that these are the same tables shown in Figure 2a.

	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
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_bool" argument can be set to TRUE if dilutions are to be included in the output. 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). The user can supply a delimiter using the "sep" function. The output of this function is critical as an argument in the "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 = "_"
)
```

Table 4: Well locations of each sample. Included in this is the dilution factor which has been transformed as the negative log of original dilution values. This table is important for input into the "plate_view" function.

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)
```

Table 5: Metadata which is included in the header of the Excel file.

Meta_ID	Meta_info
User	USER
Path	C:/Program Files (x86)/BMG/Omega/User/Data
Test ID	183
Test Name	RT-QuIC Plate Mode
Date	7/22/2024
Time	12:31:15 PM
ID1	20240722_r1_GR
Fluorescence (FI)	NA

Retrieving and Manipulating Raw Data

The raw, real-time data is typically found on the second sheet of the Excel workbook exported from MARS. 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.

Table 6: Raw data retrieved directly from the MARS output file.

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

Time	positive_control_p	sample_x1	sample_x3	sample_x5	sample_x7	sample_x9	sample_x11
0	748	749	740	726	720	718	675
0.75	503	501	498	508	501	509	458
1.5	564	537	515	533	523	526	460
2.25	601	555	531	557	521	524	452
3	604	559	515	557	524	523	458
3.75	626	565	525	562	516	524	452
4.5	634	560	521	565	512	518	439
5.25	635	551	515	571	507	518	448
6	654	572	513	567	511	515	439
6.75	652	568	511	572	501	519	432
7.5	652	566	501	573	507	516	435

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()
```

Table 8: Transposed raw data. This converted the data columns to rows and the rows to columns.

Sample IDs	0	0.75	1.5	2.25	3	3.75	4.5	5.25	6	6.75	7.5
positive_control_p	748	503	564	601	604	626	634	635	654	652	652
sample_x1	749	501	537	555	559	565	560	551	572	568	566
sample_x3	740	498	515	531	515	525	521	515	513	511	501
sample_x5	726	508	533	557	557	562	565	571	567	572	573
sample_x7	720	501	523	521	524	516	512	507	511	501	507
sample_x9	718	509	526	524	523	524	518	518	515	519	516
sample_x11	675	458	460	452	458	452	439	448	439	432	435

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)
```

Table 9: Normalized raw data Note that the "transposed" argument was set to false, so a call was made to the "transpose_real" function.

Sample IDs	0	0.75	1.5	2.25	3	3.75	4.5	5.25	6	6.75	7.5
positive_control_p	1.24	0.84	0.94	1	1.00	1.04	1.05	1.06	1.09	1.08	1.08
sample_x1	1.35	0.90	0.97	1	1.01	1.02	1.01	0.99	1.03	1.02	1.02
sample_x3	1.39	0.94	0.97	1	0.97	0.99	0.98	0.97	0.97	0.96	0.94
sample_x5	1.30	0.91	0.96	1	1.00	1.01	1.01	1.03	1.02	1.03	1.03
sample_x7	1.38	0.96	1.00	1	1.01	0.99	0.98	0.97	0.98	0.96	0.97
sample_x9	1.37	0.97	1.00	1	1.00	1.00	0.99	0.99	0.98	0.99	0.98
sample_x11	1.49	1.01	1.02	1	1.01	1.00	0.97	0.99	0.97	0.96	0.96

Calculations

There are three analytical metrics with dedicated functions: time-to-threshold (TtT), maxpoint 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 10 for an example of the output of these functions.

Thresholds

Many publications have different methods of determining thresholds. By convention, the most popular method is to take the average background fluorescence of the every well and add some multiple of standard deviations (Rowden et al. 2023). 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] 672.3326
```

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.

```r
df_norm <- get_real(file)[[1]] |>
  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 10: Calculated metrics. Each row indicates an individual well. This table shows only the first 12 wells which correspond to the top row of the microplate.

Sample IDs	Dilutions	MPR	MS	TtT	RAF
P	-3	5.73	1.869	16.76	0.0597
S01	-3	1.03	0.199	45.75	0.0219
S02	-3	14.30	4.951	9.14	0.1094
S03	-3	1.03	0.174	45.75	0.0219
S04	-3	15.62	6.523	12.75	0.0784
S05	-3	20.56	8.646	13.08	0.0765
S06	-3	1.02	0.219	45.75	0.0219
S07	-3	1.03	0.185	45.75	0.0219
S08	-3	1.01	0.196	45.75	0.0219
S09	-3	1.02	0.165	45.75	0.0219
S10	-3	1.08	0.151	45.75	0.0219
S11	-3	1.03	0.181	45.75	0.0219

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)
```

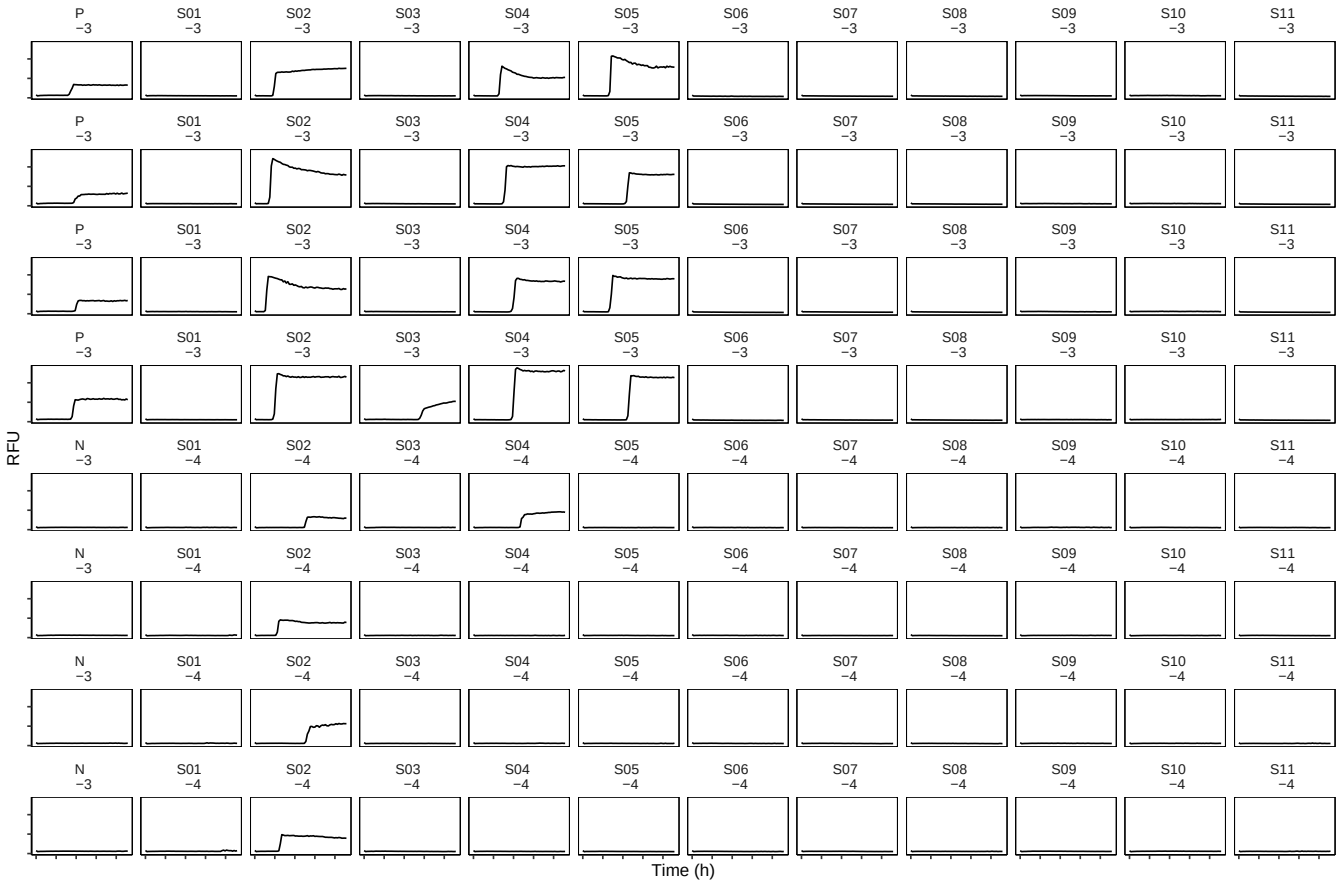
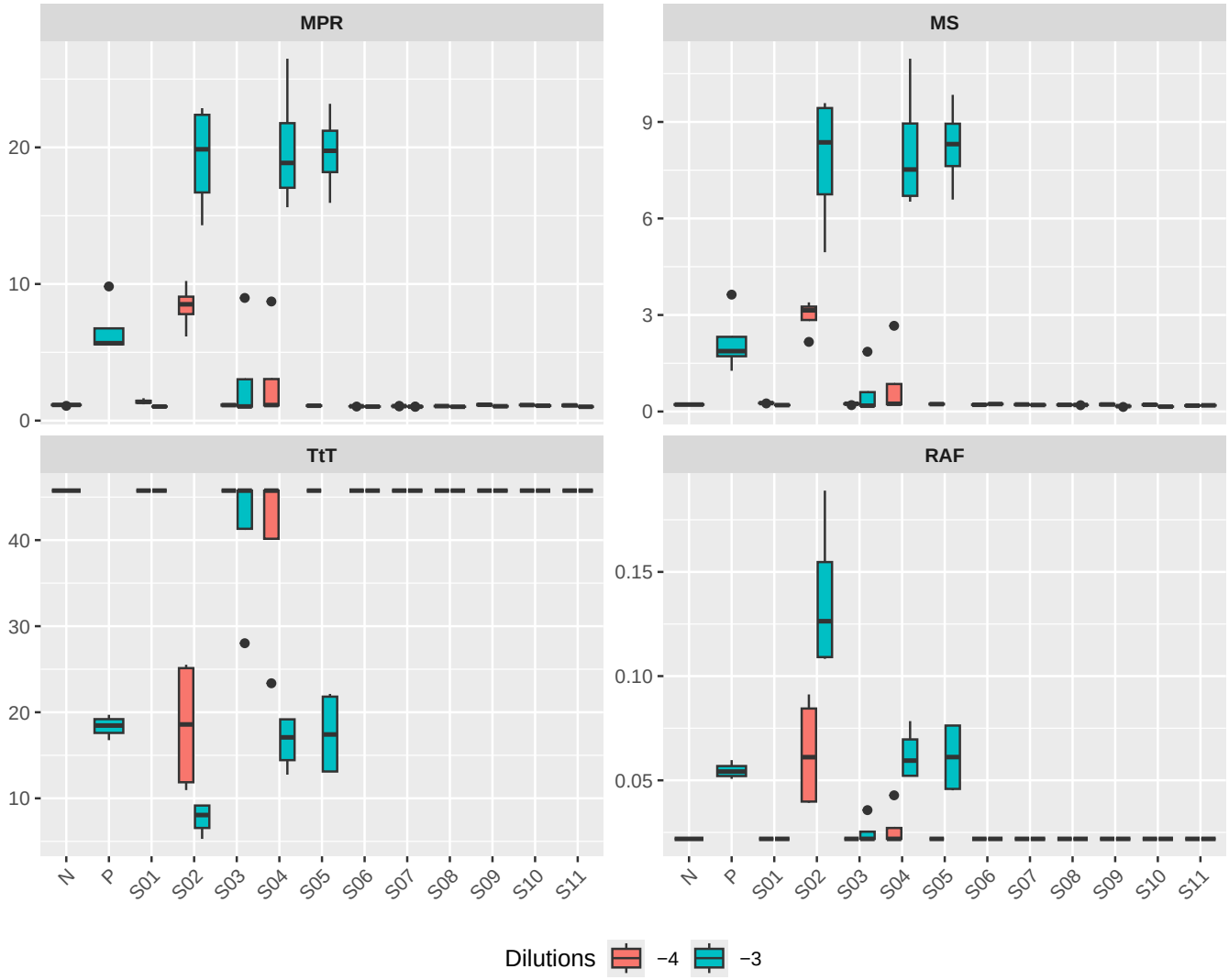


Figure 5: Plate-view analog of a 96-well microplate. Each facet in the 8x12 grid shows the real-time curves of an RT-QuIC reaction. The numbers underneath the sample IDs are the dilution factors.

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_y") +
    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")
```

Example Workflow

```
library(quicR)

# Import and Format Data ---------------------------------------------------

file <- "example.xlsx"
raw <- get_real(file)[[1]]
normal <- normalize_RFU(raw, transposed = FALSE)
meta <- file |>
  organize_tables() |>
  convert_tables()

# Calculate Key Metrics ----------------------------------------------------

mpr <- calculate_MPR(normal, data_is_norm = TRUE)
ms  <- calculate_MS(normal, data_is_norm = TRUE)
ttt <- calculate_TtT(normal, threshold = 2)

analyzed <- data.frame(
  `Sample IDs`                = meta$`Sample IDs`,
  `Dilutions`                 = meta$Dilutions,
  `Maxpoint Ratio`            = mpr,
  `Max Slope`                 = ms,
  `Time-to-Threshold`         = ttt,
  `Rate of Amyloid Formation` = 1 / ttt
)
```

Discussion

The quicR package represents a significant advancement in the standardization and reproducibility of RT-QuIC data analysis. By integrating key metrics such as time-to-threshold (TtT), maxpoint ratio (MPR), and maximum slope (MS), quicR addresses critical gaps in the field, providing researchers and diagnosticians with a robust toolkit for interpreting complex fluorescence data.

One of the primary strengths of quicR lies in its flexibility and user-centric design. The package leverages R's powerful ecosystem, including the tidyverse and ggplot2, to streamline workflows and create high-quality, customizable visualizations. This ensures accessibility for a wide range of users, from experienced data scientists to wet-lab researchers unfamiliar with programming. Additionally, the incorporation of open-source principles allows the broader scientific community to contribute to its development, fostering innovation and adaptability.

Despite these strengths, there are limitations to consider. Currently, quicR is tailored to data exported from the MARS software, which may limit its applicability to researchers using alternative fluorescence readers. Future iterations of the package could expand compatibility by incorporating functions to handle diverse data formats. Furthermore, while quicR includes robust visualization tools, users seeking highly specialized plots may require additional customization beyond the package's default capabilities.

Another avenue for improvement lies in the standardization of RT-QuIC diagnostic criteria. quicR provides tools to calculate key metrics, but consensus on thresholds and interpretations remains a challenge for the field. Collaborative efforts among researchers and clinicians are necessary to define universal criteria, enabling quicR to fully realize its potential as a diagnostic aid. Diagnostic determinations could easily be built into the library, but a larger consensus within the research community will need to be reached to warrant inclusion.

Conclusion

quicR offers a powerful solution for the cleaning, analysis, and visualization of RT-QuIC data, addressing critical needs in a rapidly evolving field. By enabling consistent data handling and interpretation, quicR lays the groundwork for improved diagnostic consistency and reproducibility. The package's open-source nature ensures that it will continue to evolve, integrating new insights and technologies as they emerge.

As RT-QuIC technology advances, tools like quicR will play a pivotal role in bridging the gap between assay development and practical application. By equipping researchers with reliable, standardized tools, quicR not only supports the study of prion and protein misfolding disorders but also serves as a model for the development of software solutions in other diagnostic fields.

Acknowledgments

Special thanks to Beni Altmann at The Comprehensive R Archive Network (CRAN) for help during the submission process to CRAN. We thank Tiffany Wolf and Marc Schwabenlander for their support through the Minnesota Center for Prion Research and Outreach. We would like to acknowledge Suzanne Stone and Sarah Gresch for maintaining lab operations.

Funding

This work was funded through the Legislative-Citizen Commission on Minnesota Resources (LCCMR).

References

Atarashi, Ryuichiro, Kazunori Sano, Katsuya Satoh, and Noriyuki Nishida. 2011. "Real-Time Quaking-Induced Conversion: A Highly Sensitive Assay for Prion Detection." *Prion* 5 (July): 150–53. https://doi.org/10.4161/pri.5.3.16893.

Bongianni, Matilde, Anna Ladogana, Stefano Capaldi, Sigrid Klotz, Simone Baiardi, Annachiara Cagnin, Daniela Perra, et al. 2019. "α-Synuclein RT-QuIC Assay in Cerebrospinal Fluid of Patients with Dementia with Lewy Bodies." *Annals of Clinical and Translational Neurology* 6 (10): 2120–26.

Christenson, Peter R., Manci Li, Gage Rowden, Peter A. Larsen, and Sang-Hyun Oh. 2023. "Nanoparticle-Enhanced RT-QuIC (Nano-QuIC) Diagnostic Assay for Misfolded Proteins." *Nano Letters* 23 (9): 4074–81. https://doi.org/10.1021/acs.nanolett.3c01001.

Dassanayake, Rohana P, Christina D Orrú, Andrew G Hughson, Byron Caughey, Telmo Graça, Dongyue Zhuang, Sally A Madsen-Bouterse, Donald P Knowles, and David A Schneider. 2016. "Sensitive and Specific Detection of Classical Scrapie Prions in the Brains of Goats by Real-Time Quaking-Induced Conversion." *J. Gen. Virol.* 97 (3): 803–12.

Fiorini, Michele, Giorgia Iselle, Daniela Perra, Matilde Bongianni, Stefano Capaldi, Luca Sacchetto, Sergio Ferrari, et al. 2020. "High Diagnostic Accuracy of RT-QuIC Assay in a Prospective Study of Patients with Suspected sCJD." *International Journal of Molecular Sciences* 21 (3). https://doi.org/10.3390/ijms21030880.

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

Franceschini, Alessia, Simone Baiardi, Andrew G. Hughson, Neil McKenzie, Fabio Moda, Marcello Rossi, Sabina Capellari, et al. 2017. "High Diagnostic Value of Second Generation CSF RT-QuIC Across the Wide Spectrum of CJD Prions." *Sci Rep*, September. https://doi.org/10.1038/s41598-017-10922-w.

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.

Groveman, Bradley R, Christina D Orrù, Andrew G Hughson, Lynne D Raymond, Gianluigi Zanusso, Bernardino Ghetti, Katrina J Campbell, Jiri Safar, Douglas Galasko, and Byron Caughey. 2018. "Rapid and Ultra-Sensitive Quantitation of Disease-Associated α-Synuclein Seeds in Brain and Cerebrospinal Fluid by αSyn RT-QuIC." *Acta Neuropathologica Communications* 6 (1): 7.

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.

Holz, Carine L, Joseph R Darish, Kelly Straka, Nicole Grosjean, Steven Bolin, Matti Kiupel, and Srinand Sreevatsan. 2021. "Evaluation of Real-Time Quaking-Induced Conversion, ELISA, and Immunohistochemistry for Chronic Wasting Disease Diagnosis." *Front Vet Sci* 8: 824815.

Hwang, Soyoun, M Heather West Greenlee, Anne Balkema-Buschmann, Martin H Groschup, Eric M Nicholson, and Justin J Greenlee. 2018. "Real-Time Quaking-Induced Conversion Detection of Bovine Spongiform Encephalopathy Prions in a Subclinical Steer." *Frontiers in Veterinary Science* 4 (JAN): 19.

Lee, Dong Jun, Peter R. Christenson, Gage Rowden, Nathan C. Lindquist, Peter A. Larsen, and Sang-Hyun

Oh. 2024. “Rapid on-Site Amplification and Visual Detection of Misfolded Proteins via Microfluidic Quaking-Induced Conversion (Micro-QuIC).” *Npj Biosensing* 1 (6). https://doi.org/10.1038/s44328-024-00006-x.

Metrick, Michael A, 2nd, Natália do Carmo Ferreira, Eri Saijo, Allison Kraus, Kathy Newell, Gianluigi Zanusso, Michele Vendruscolo, Bernardino Ghetti, and Byron Caughey. 2020. “A Single Ultrasensitive Assay for Detection and Discrimination of Tau Aggregates of Alzheimer and Pick Diseases.” *Acta Neuropathol Commun* 8 (1): 22.

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.

Orrú, Christina D., Jason M. Wilham, Lucienne D. Raymond, Fabian Kuhn, Brent Schroeder, Andreas J. Raeber, and Byron Caughey. 2012. “Prion Disease Blood Test Using Immunoprecipitation and Improved Quaking-Induced Conversion.” *mBio* 2 (3). https://doi.org/10.1128/mBio.00078-11.

Orrù, Christina D, Bradley R Groveman, Andrew G Hughson, Matteo Manca, Lynne D Raymond, Gregory J Raymond, Katrina J Campbell, Kelsie J Anson, Allison Kraus, and Byron Caughey. 2017. “RT-QuIC Assays for Prion Disease Detection and Diagnostics.” In *Methods in Molecular Biology*, 1658:185–203. Humana Press Inc.

Picasso-Risso, Catalina, Marc D Schwabenlander, Gage Rowden, Michelle Carstensen, Jason C Bartz, Peter A Larsen, and Tiffany M Wolf. 2022. “Assessment of Real-Time Quaking-Induced Conversion (RT-QuIC) Assay, Immunohistochemistry and ELISA for Detection of Chronic Wasting Disease Under Field Conditions in White-Tailed Deer: A Bayesian Approach.” *Pathogens* 11 (5).

R Core Team. 2024. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. https://www.R-project.org/.

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.

Wilham, Jason M, Christina D Orrú, Richard A Bessen, Ryuichiro Atarashi, Kazunori Sano, Brent Race, Kimberly D Meade-White, Lara M Taubner, Andrew Timmes, and Byron Caughey. 2010. “Rapid End-Point Quantitation of Prion Seeding Activity with Sensitivity Comparable to Bioassays.” *PLOS Pathog* 6 (December). https://doi.org/10.1371/journal.ppat.1001217.

Appendix