

Quantitative Growth Curve Evaluation with Package **QurvE**

Nicolas T. Wirth

2022-10-06

Please note: This vignette will be updated from time to time when new features are implemented. Please find the most recent version at the **QurvE** GitHub repository.

1 Introduction

In virtually all disciplines of biology dealing with living organisms, from classical microbiology to applied biotechnology, it is routine to characterize the **growth** of the species under study. **QurvE** provides a suite of analysis tools to make such growth profiling quick, efficient, and reproducible. In addition, it allows the characterization of **fluorescence** data for, e.g., biosensor characterization in plate reader experiments (further discussed in the vignette *Quantitative Fluorescence Curve Evaluation with Package **QurvE***). All computational steps to obtain an in-depth characterization are combined into user-friendly *workflow* functions and a range of plotting functions allow the visualization of fits and the comparison of organism performances.

Any physiological parameter calculated (e.g., growth rate μ , doubling time t_D , lag time λ , density increase ΔY , or equivalent fluorescence parameters) can be used to perform a dose-response analysis to determine the *half-maximal effective concentration* (EC_{50}).

The package is build on the foundation of the two R packages from Kahm *et al.* (2010) and Petzoldt (2022). **QurvE** was designed to be usable with minimal prior knowledge of the R programming language or programming in general. You will need to be familiar with the idea of running commands from a console or writing basic scripts. For R beginners, this is a great starting point, there are some good resources here and we suggest using the RStudio application. It provides an environment for writing and running R code.

With consideration for R novices, **QurvE** establishes a framework in which a complete, detailed growth curve analysis can be performed in two simple steps:

1. *Read data* in custom format or parse data from a plate reader experiment.
2. *Run workflow*, including fitting of growth curves, dose-response analysis, and rendering a report that summarizes the results.

All computational results of a workflow are stored in a data container (list) and can be visualized by passing them to the generic function `plot(list_object)`. **QurvE** further extends the user's control over the fits by defining thresholds and quality criteria, allows the direct parsing of data from plate reader result files, and calculates parameters for an additional growth phase (bi-phasic growth).

2 Installation

2.1 Development version

Install the must current version with package **devtools**:

```
install.packages("devtools")
library(devtools)
install_github("NicWir/QuurvE")
```

3 Growth profiling methods

Three methods are available to characterize growth curves:

1. Fit parametric growth models to (log-transformed) growth data
2. Determine maximum growth rates (μ_{\max}) from the log-linear part of a growth curve using a heuristic approach proposed as the “growth rates made easy”-method by Hall *et al.* (2014). Do so, **QuurvE** uses code from the package Petzoldt (2022), but adds user-defined thresholds for (i) R^2 values of linear fits, (ii) relative standard deviations (RSD) of estimates slopes, and (iii) the minimum fraction of total density increase (ΔY) a regression window should cover to be considered for the analysis. These thresholds ensure a more robust and reproducible identification of the linear range that best describes the growth curve. Additionally, parameters for a *secondary growth phase* can be extracted for bi-linear growth curves.¹

The algorithm works as follows:

- i. Fit linear regressions [with the Theil-Sen estimator (Sen, 1968; Theil, 1992)] to all subsets of **h** consecutive, log-transformed data points (sliding window of size **h**). If, for example, **h=5**, fit a linear regression to points 1 ... 5, 2 ... 6, 3 ... 7 and so forth.
- ii. Find the subset with the highest slope μ_{\max} . Do the R^2 and RSD values of the regression meet the defined thresholds and do the data points within the regression window account for at least a defined fraction of the total density increase? If not, evaluate the regression with the second highest slope, and so forth.
- iii. Include also the data points of adjacent subsets that have a slope of at least a *definedquota* $\times \mu_{\max}$, e.g., all regression windows that have at least 95% of the maximum slope.
- iv. Fit a new linear model to the extended data window identified in step iii.

If **biphasic = TRUE**, the following steps are performed to define a second growth phase:

- i. Perform a smooth spline fit on the data with a smoothing factor of 0.5.
 - ii. Calculate the second derivative of the spline fit and perform a smooth spline fit of the derivative with a smoothing factor of 0.4.
 - iii. Determine local maxima and minima in the second derivative.
 - iv. Find the local minimum following μ_{\max} and repeat the heuristic linear method for later time values.
 - v. Find the local maximum before μ_{\max} and repeat the heuristic linear method for earlier time values.
 - vi. Choose the greater of the two independently determined slopes as μ_{\max}^2 .
3. Perform a smooth spline fit on (log-transformed) growth data and extract μ_{\max} as the maximum value of the first derivative¹.

If **biphasic = TRUE**, the following steps are performed to define a second growth phase:

¹For linear and nonparametric (i.e, smooth spline) fits, the lag time is calculated as the intersect of the tangent at μ_{\max} and a horizontal line at the first data point (y_0). For the analysis of bi-phasic growth curves, the lag time is extracted as the lower of the two obtained λ values.

- i. Determine local minima within the first derivative of the smooth spline fit.
- ii. Remove the ‘peak’ containing the highest value of the first derivative (i.e., μ_{max}) that is flanked by two local minima.
- iii. Repeat the smooth spline fit and identification of maximum slope for later time values than the local minimum after μ_{max} .
- iv. Repeat the smooth spline fit and identification of maximum slope for earlier time values than the local minimum before μ_{max} .
- v. Choose the greater of the two independently determined slopes as μ_{max} ².

4 Data formats

QurvE accepts files with the formats *.xls*, *.xlsx*, *.csv*, *.tsv*, and *.txt* (tab separated). The data in the files should be structured as shown in Figure ???. Alternatively, data parsers are available that allow direct import of raw data from different culture instruments. For a list of currently supported devices, please run `?parse_data`.

4.1 Custom format

To ensure compatibility with any type of measurement and data type (e.g., optical density, cell count, measured dimensions), QurvE uses a custom data layout. Here the first column contains *time values* and ‘Time’ in the top cell, cells #2 and #3 are ignored. The remaining columns contain *measurement values* and the following sample identifiers in the top three rows:

1. Sample name; usually a combination of organism and condition, or ‘blank’.
2. Replicate number; replicates are identified by *identical* names and concentration values. If only one type of replicate (biological or technical) was performed, enter numerical values here. If **both** biological and technical replicates of these biological replicates have been performed, the biological replicates shall be indicated with numbers and the technical replicates with letters. The technical replicates are then combined with their average value.
3. (*optional*) Concentration values of an added compound; this information is used to perform a dose-response analysis.

Several experiments (e.g., runs on different plate readers) can be combined into a single file and analyzed simultaneously. Therefore, different experiments are marked by the presence of separate time columns. Different lengths and values in these time columns are permitted.

To read data in custom format, run:

```
input <- read_data(data.density = 'path_to_data_file',
                  csvsep = ';', # or ',',
                  dec = '.', # or ',',
                  sheet.density = 1, # specify the number (or "name") of the .xls/.xlsx file sheet containing the data,
                  subtract.blank = TRUE,
                  calibration = NULL
)
```

The ‘**data.density**’ argument takes the path to the file containing experimental data in custom format. ‘**csvsep**’ specifies the separator symbol (only required for *.csv* files; default: ‘;’). ‘**dec**’ is the decimal separator (only required for *.csv*, *.tsv*, or *.txt* files; default: ‘.’). If an Excel file format is used, ‘**sheet.density**’ specifies the number or name (in quotes) of the sheet containing the data. If ‘**subtract.blank = TRUE**’, columns with name ‘blank’ will be combined by their row-wise average, and the mean values will be subtracted from the measurements of all remaining samples. For the ‘**calibration**’ argument, a formula can be provided in the form ‘*y = function(x)*’ (e.g., ‘*y = x * 2 + 0.5*’) to transform all measurement values.