

## The population growth rate

... is a measure of fitness (→ Fig. 1)

- Applicable to animals, plants, bacteria
- Dependent on resources (nutrients)
- Affected by physical environment (high temperatures) and toxic substances (micro-pollutants, pharmaceuticals, antibiotics)

## Database like data structure (→ Fig. 2)

- Flexible, tabular data structure, independent of specific experimental setup and application area
- The package includes real data for testing:
  - An almost ideal and nice data set: **antibiotic**
  - A typical challenging data set: **bactgrowth**

## Included growth models (→ Fig. 3)

- "growthrates made easy"-approach (Hall et al. 2014, MolBEnv)
  - Smoothing splines (Kahm et al. 2010, JSS)
  - Parametric models (= nonlinear regression)
    - Popular growth curves built-in (exponential, logistic, Baranyi, ...)
    - Differential equation models (numerically solved with deSolve)
    - User-defined growth models
- User-defined models can be written in **R** or with packages **cOde** (inline **C**) or **rodeo** (**Fortran**) to make it faster

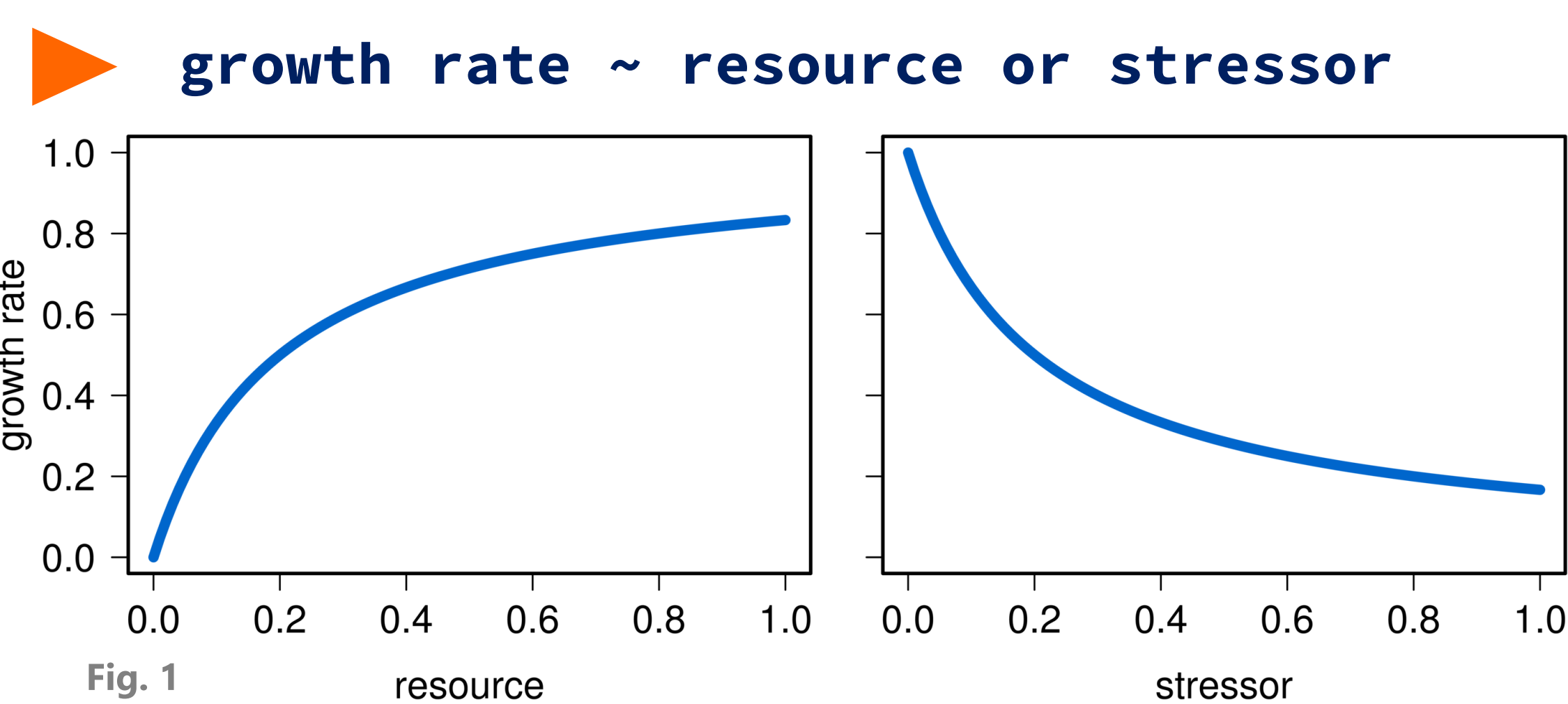


Fig. 1

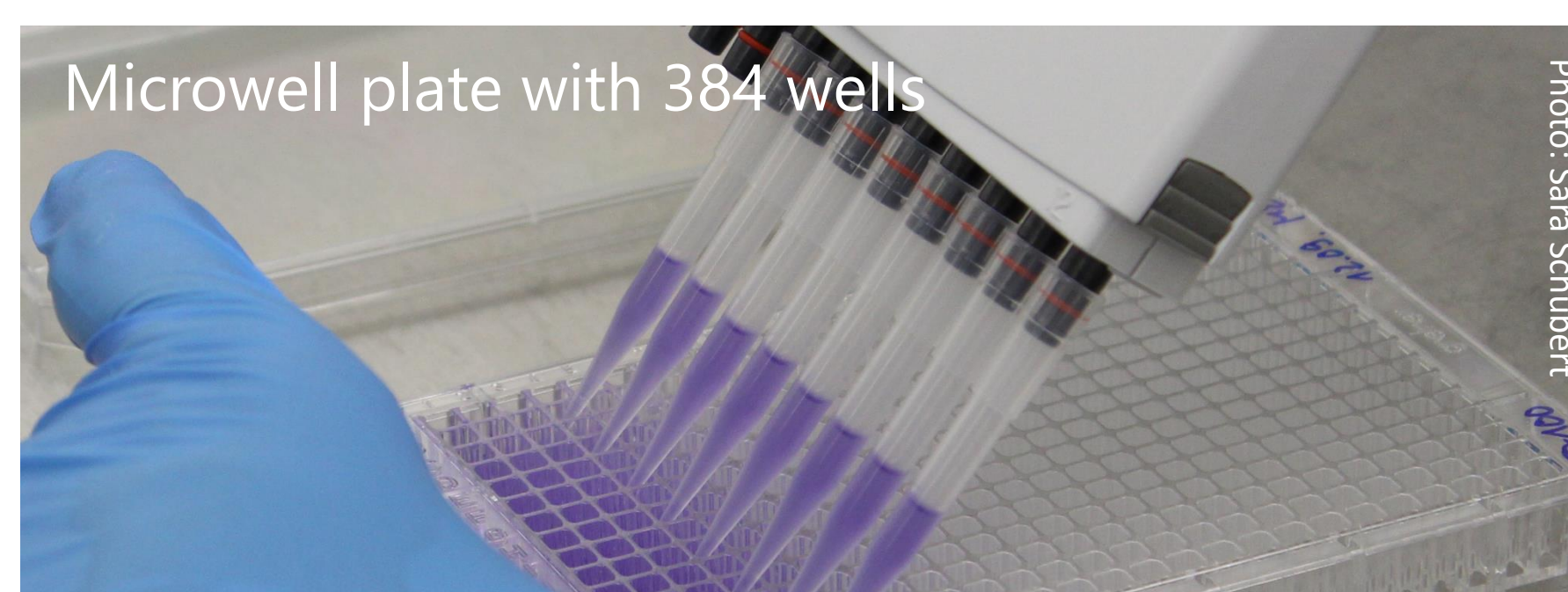


Fig. 2:  
Tabular data  
structure

time	strain	conc	repl	value
0.0	R	0.0	R3	0.0088
0.5	R	0.0	R3	0.0098
1.0	R	0.0	R3	0.0128
1.5	R	0.0	R3	0.0178
2.0	R	0.0	R3	0.0228
...	...	...	...	...
29.5	T	2.5	R6	0.0166
30.0	T	2.5	R6	0.0166

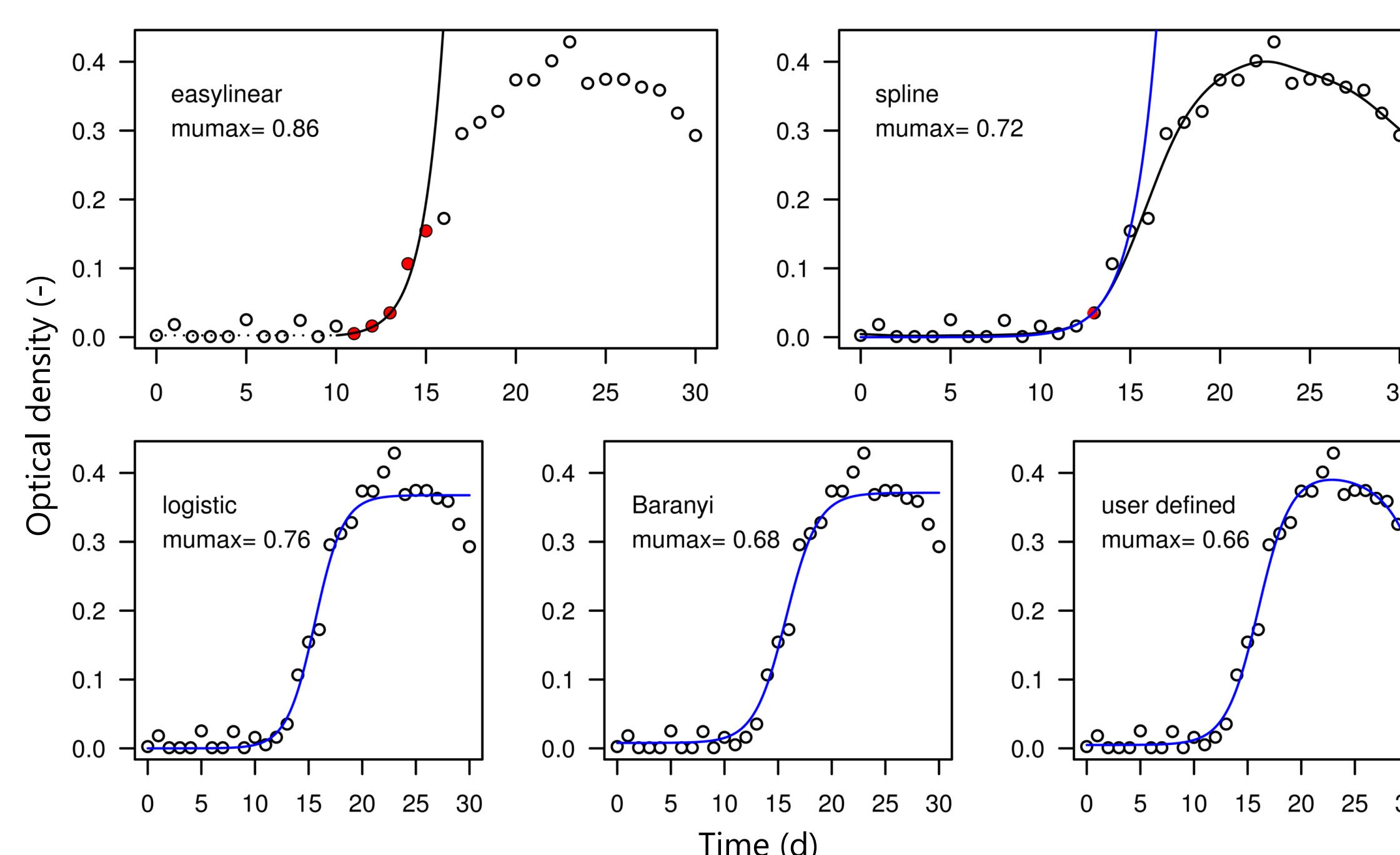


Fig. 3: Different approaches and models for growth rate estimation

## Why another R package?

- Large amounts of data
- Proprietary software: high manual effort for data import/export, difficult to automate
- R, Matlab, Python: "Not all biologists want to spend the rest of their lives as programmers" (Patton, 2005)

## Design Goals

- Software package that hides most (but not all) programming behind the scenes
- Object oriented and user-extensible
- Flexible, tabular data structure
- Results as tables, quick graphics

## Download and Installation

- Free and Open Source Software (GPL 2 or 3)
- Windows, Linux, MacOS
- Available from CRAN (<https://cran.r-project.org>)
- Development on Github

<http://www.github.com/tpetzoldt/growthrates>



## Documentation

- Open Access: online and part of the package
- Interactive web-apps

<http://limno-live.hydro.tu-dresden.de>

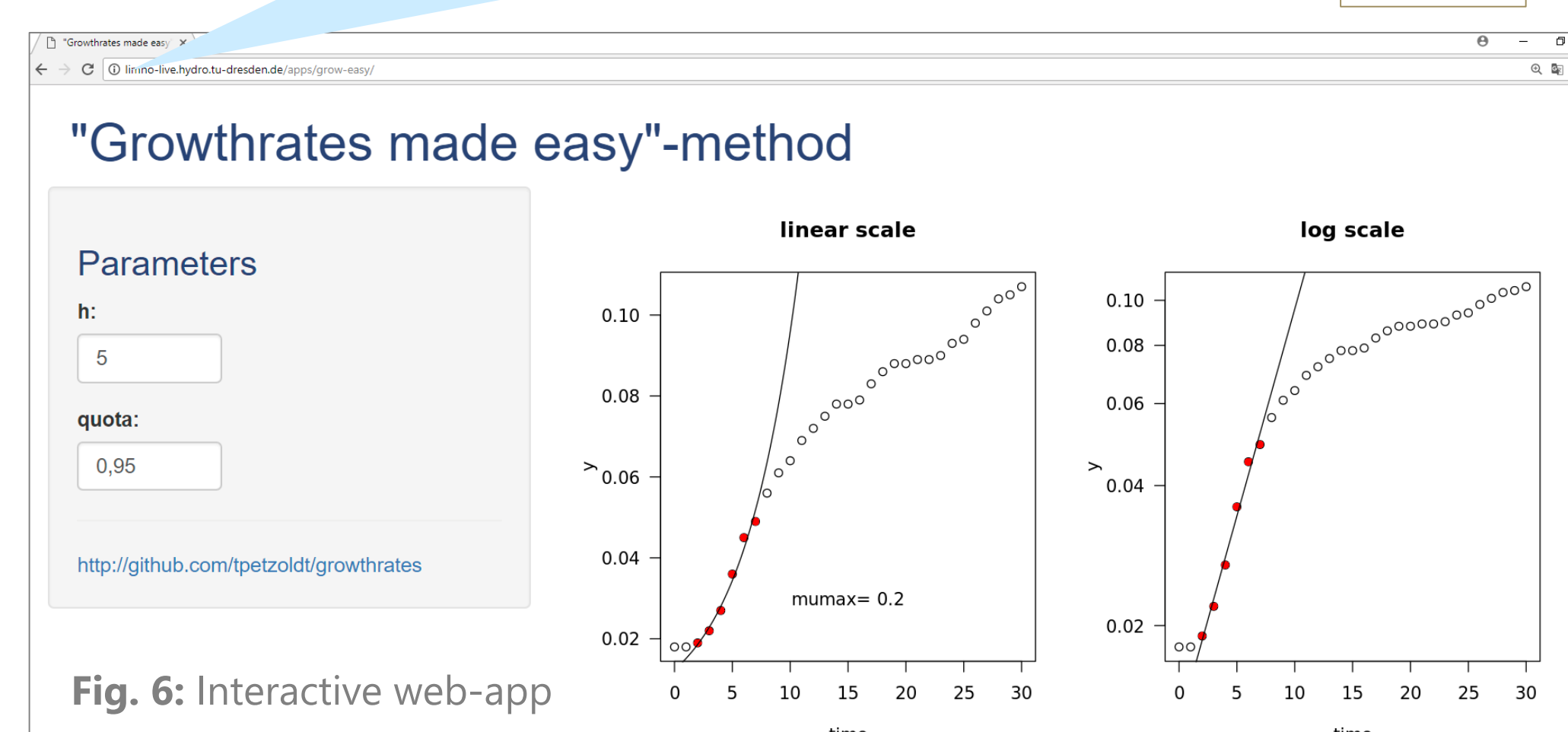


Fig. 6: Interactive web-app

## Data

## Fit single growth curve to data subset

- Find most suitable model for your purpose
- Compare different approaches
- Select best start parameters ...

→ Fig. 3

## Fit complete experiment in one rush

- Formula interface to split data into subsets → Fig. 4
  - Makes use of generic data base structure
  - No loop programming necessary
- Full availability of standard R features: calculations, subsetting, exclusion of data
- Select fit parameters, leave the remaining fixed
- Parallel computing on multicore processors
- Plot complete set of curves with a single command
- Extract results in tabular form

## Fit dose-response curve → Fig. 5

- Compatible to vast diversity of R's data analysis, plotting functions and add-on packages
- Dose response curves with **nls** or package **drc**
- Confidence intervals with package **propagate**

## Use the model

- Use the selected model structure and parameters for nonlinear mixed effect modeling with package **lme4**
- Implement the dose response curve as a submodel in complex mechanistic models

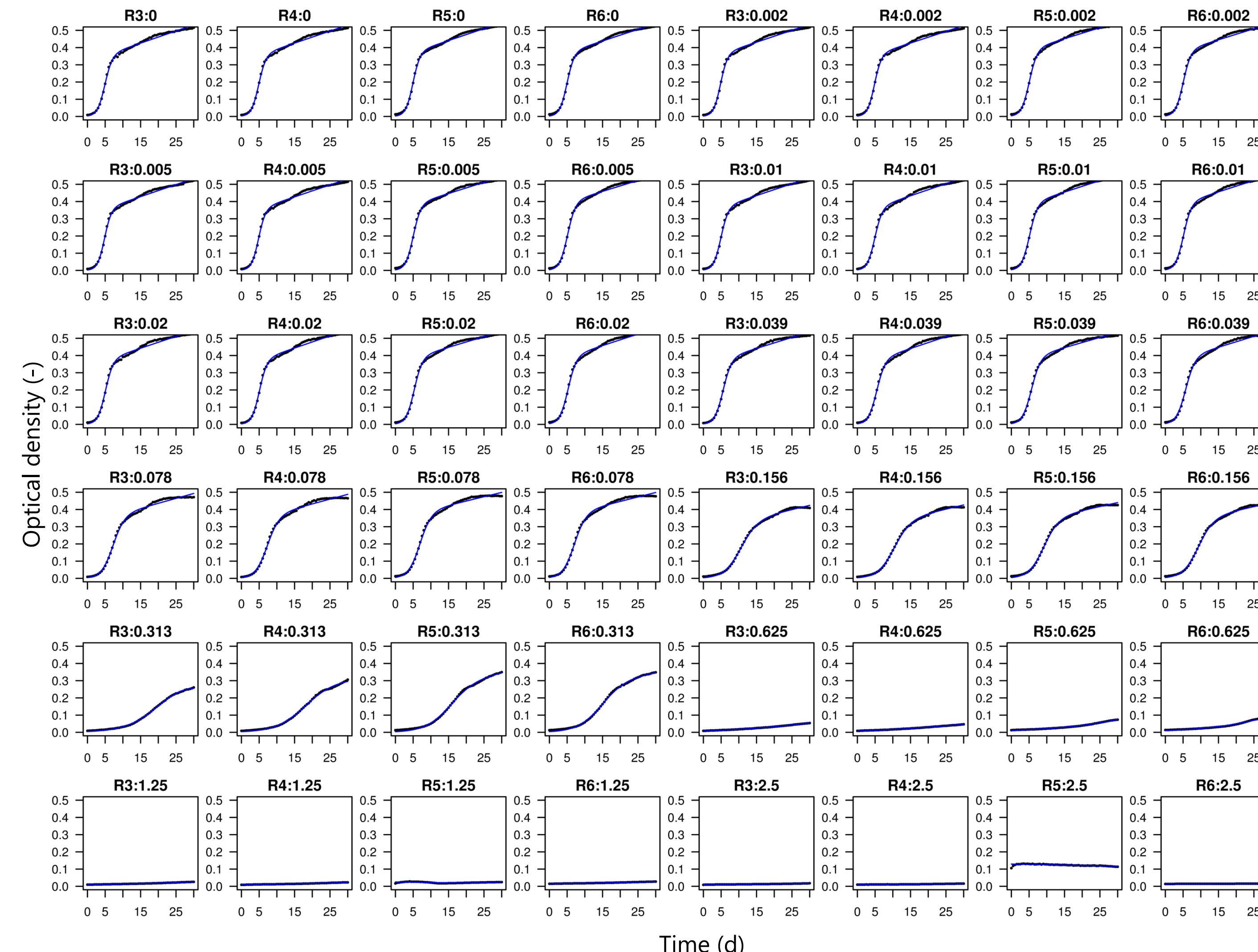


Fig. 4: Fit of a complete experiment, **value ~ grow\_userdefined(time, parms) | repl + conc**

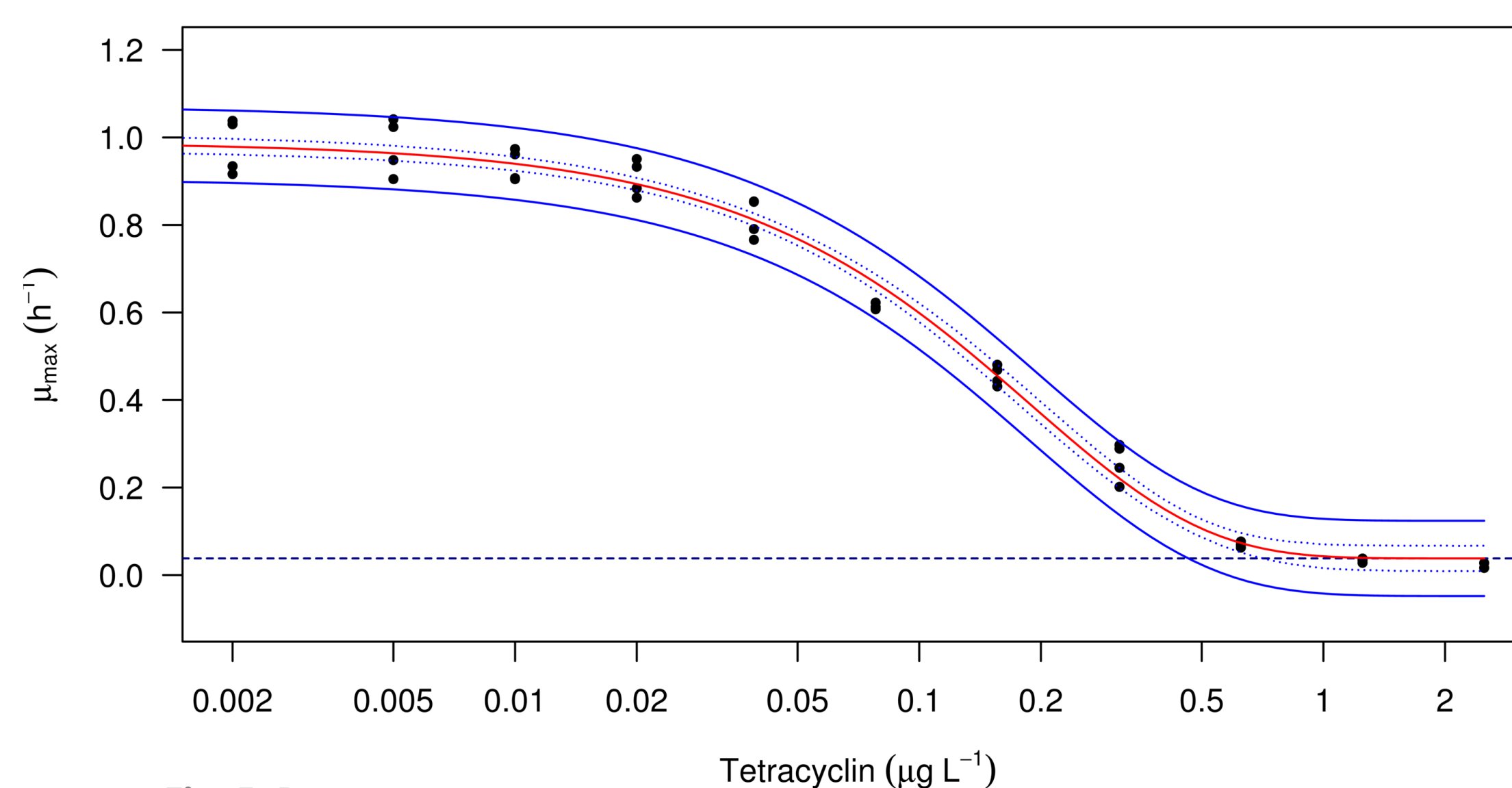


Fig. 5: Dose-response curve

## Code Examples

```
## load package and example data
library(growthrates)
data(antibiotic)

## load data, split table, select a single example
samples <- multisplit(value ~ time|conc:repl,
  data = antibiotic)
dat <- samples[["0.078:R3"]]

## "growth rates made easy"-method
fit1 <- fit_easylinear(dat$time, dat$value)
plot(fit1, log="y")
summary(fit1)

## smoothing spline method
fit2 <- fit_spline(dat$time, dat$value, spar=0.5)
plot(fit2, log="y")

## parametric model
parms <- c(y0=0.01, mumax=0.2, K=0.5)
fit3 <- fit_growthmodel(grow_logistic, parms,
  dat$time, dat$value)
plot(fit3)

## a user-defined growth model
grow_userdef <- function(time, parms) {
  with(as.list(parms), {
    y <- ((K + dK*time) * y0) /
      (y0 + ((K + dK*time) - y0) * exp(-mumax * time))
    data.frame(time = time, y = y, log_y = log(y))
  })
}

## fit complete series of experiments in one rush
parms <- c(y0=0.01, mumax=0.2, K=0.5, dK=0.01)
fits <- all_growthmodels(
  value ~ grow_userdef(time, parms) | repl + conc,
  data = antibiotic, p = parms, nccores = 4)

## plot all figures
par(mfrow=c(6,8), mar=c(2.5, 2.5, 1.5, 0), las=1)
plot(fits, ylim=c(0, 0.6), cex = 0.7)
results(fits)
```

## Conclusion

Package **growthrates** is lightweight and open:

- Avoid (the complicated) parts of programming
  - Export result tables to your favorite data analysis and graphics software
  - Integrate growth rate fitting in your pipeline
- Reproducible research with citable methods