



TECHNISCHE
UNIVERSITÄT
DRESDEN

Maximum Growth Rates Estimation

with

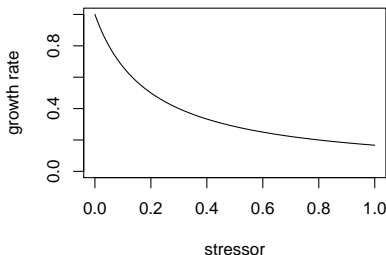
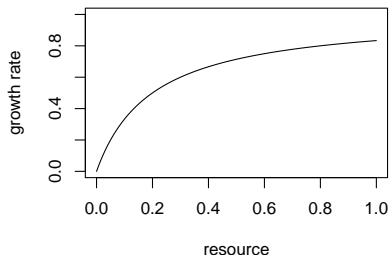
-package **growthrates**

Thomas Petzoldt, David Kneis and Claudia Seiler

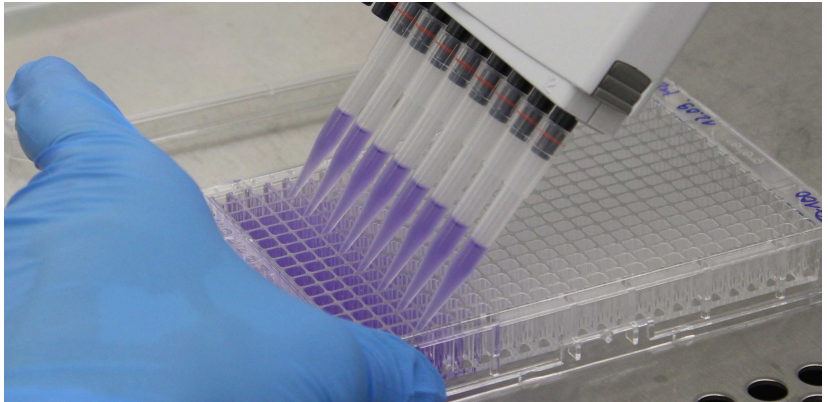
Dresden University of Technology, Faculty of Environmental Sciences, Dresden (Germany)

Motivation

- ▶ The growth rate of a population is a measure of fitness.
- ▶ applicable to animals, plants, bacteria
- ▶ test substances / model organisms
 - ▶ resources (nutrients)
 - ▶ toxic substances (micro-pollutants, pharmaceuticals, antibiotics)
 - ▶ physical environment (high temperatures)
- ▶ growth rate \sim resource or stressor



Microwell plate with 384 wells



- ▶ modern biology → large amounts of data
- ▶ data analysis → bottleneck

Data analysis bottleneck

Limitations of proprietary laboratory software

- ▶ high manual effort with “user friendly” Windows or Mac programs
- ▶ data import/export between different software packages

R can do it. What’s the problem?

Not all biologists “want to spend the rest of their lives as programmers”¹:

- ▶ Babylonian diversity of parametric models
- ▶ → difficulty to compare approaches
- ▶ → automatisisation: need to fit lots of growth rates in a batch

Fitting growth rates was a FAQ for me.

¹Geoff Patton at ECOLOG-L,
<https://listserv.umd.edu/cgi-bin/wa?A2=ind0505d&L=ecolog-1&D=0&P=5326>

Create a package for USErS

Different approaches – unique user interface

- ▶ *growth rates made easy*-approach (Hall et al. 2013, MolbEv)
- ▶ smoothing spline approach (Kahm et al. 2010, JSS)
- ▶ parametric models

Generic data structure, generic methods (S3/S4 OOP)

- ▶ formula interface to fit several data sets at once
- ▶ functions to extract results
- ▶ visualization functions

Small set of built-in models – extensible by the user

- ▶ given in closed form or as system of ODEs
- ▶ → numerical integration of ODE models
- ▶ option to use compiled code

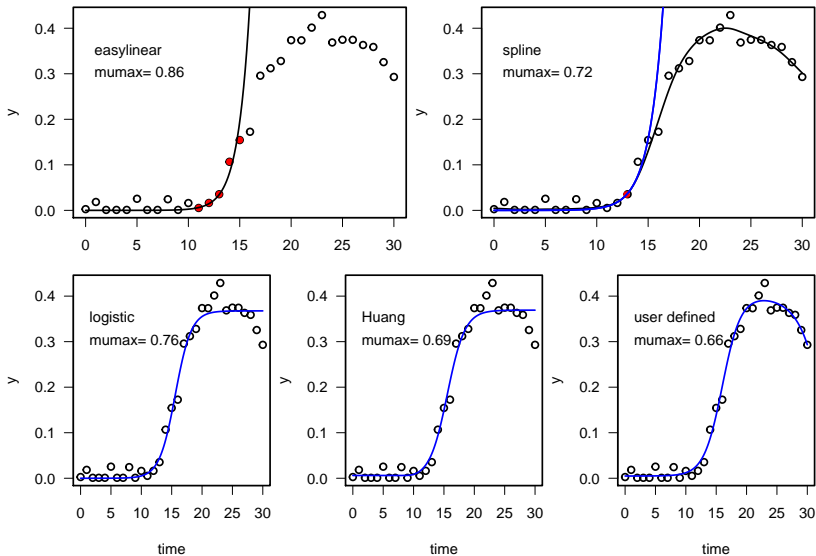
Database-like and NOT cross tables

```
library("growthrates")  
data(bactgrowth)  
head(bactgrowth)
```

strain	replicate	conc	time	value
T	2	0	0	0.013
T	2	0	1	0.014
T	2	0	2	0.017
T	2	0	3	0.022
T	2	0	4	0.030
T	2	0	5	0.039

- ▶ independent variable: time
- ▶ dependend variable: value (counts, optical density, ...)
- ▶ arbitrary number of factors (strain, replicate, treatment, ...)

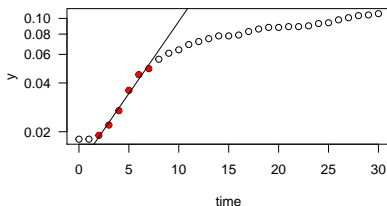
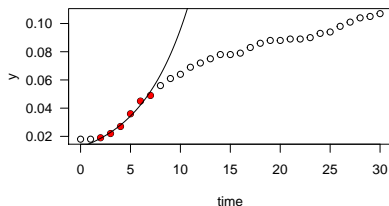
“Easylinear”, Spline, Parametric fits



Heuristic approach: “growth rates made easy”

Proposed by Hall et al. 2013

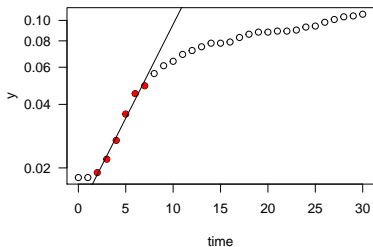
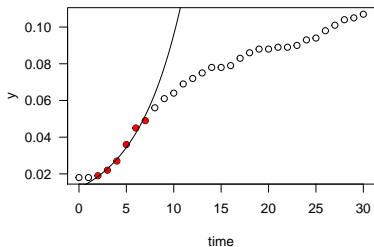
1. log transform the data
2. select all sets of 5 consecutive points
3. fit linear regressions
4. take the one with the steepest slope
5. include more points as long as slope is $\geq 95\%$ of maximum slope



Shiny Demo: <http://limno-live.hydro.tu-dresden.de/apps/grow-easy>

Heuristic approach: “growth rates made easy”

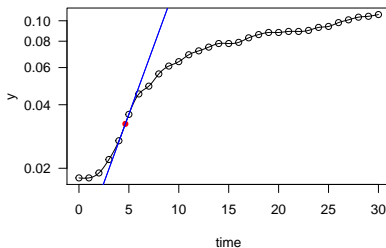
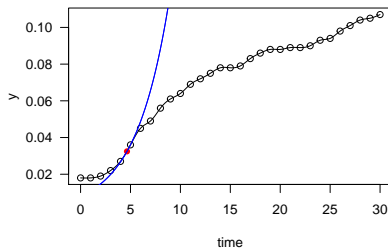
```
splitted.data <- multisplit(  
  value ~ time|strain:conc:replicate, data=bactgrowth)  
dat <- splitted.data[[1]]  
fit <- fit_easylinear(dat$time, dat$value)  
plot(fit); plot(fit, log="y")
```



- ▶ multisplit with formula interface
- ▶ plot, summary, coef, deviance, ..., results

Smoothing spline method

```
fit2 <- fit_spline(dat$time, dat$value)
plot(fit2); plot(fit2, log="y")
```



- ▶ μ_{max} = maximum of 1st derivative
- ▶ degree of smoothing:
 - ▶ cross validation (by smooth.spline)
 - ▶ fixed value spar, plate reader data are typically auto-correlated

Parametric models

In closed form

- ▶ exponential, logistic, Gompertz, Baranyi, Huang
- ▶ package **FME** for (constrained) nonlinear optimization

As differential equation (ODE)

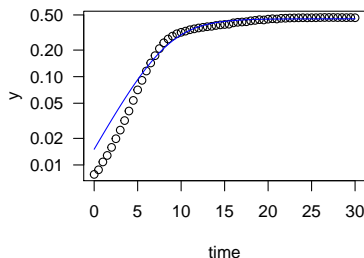
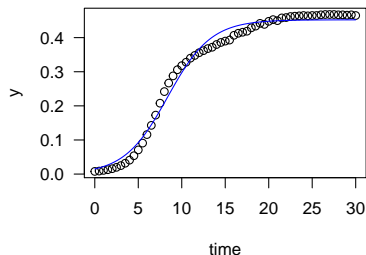
- ▶ twostep, generalized logistic
- ▶ package **deSolve** for numerical integration

User-defined

- ▶ in closed form or as ODE
- ▶ plain R or with packages **cOde** (C) or **rodeo** (Fortran)

Parametric models: An example

```
antibiotic <- readRDS("antibiotic.Rda")
dat <- subset(antibiotic, conc==0.078 & repl=="R4")
parms <- c(y0=0.01, mumax=0.2, K=0.5)
fit <- fit_growthmodel(grow_logistic, parms, dat$time, dat$value)
plot(fit); plot(fit, log="y")
```

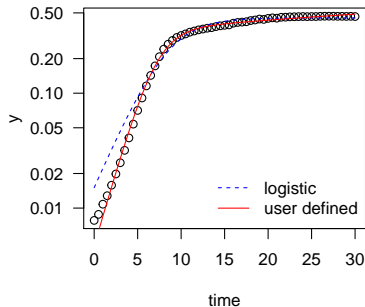
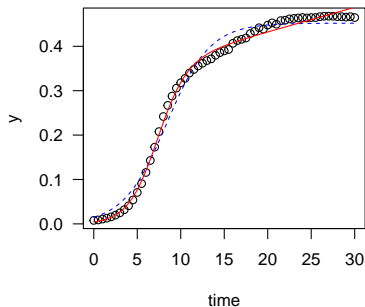


Alternatives to the standard logistic function in a [Shiny Demo](#):

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

User-defined model

```
grow_userdefined <- function(time, parms) {  
  with(as.list(parms), {  
    y <- ((K + dK*time) * y0) /  
      (y0 + ((K + dK*time) - y0) * exp(-mumax * time))  
    as.matrix(data.frame(time = time, y = y, log_y = log(y)))  
  })  
}  
parms <- c(y0=0.01, mumax=0.2, K=0.5, dK=0.01)  
fit2 <- fit_growthmodel(grow_userdefined, parms, dat$time, dat$value)
```



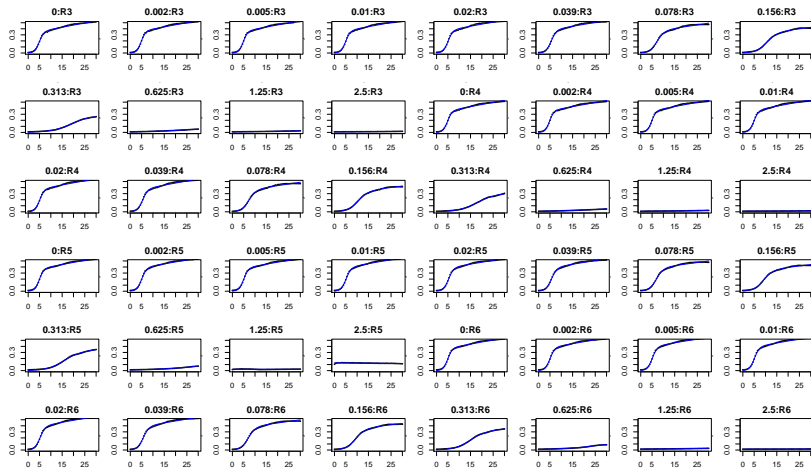
Fit a complete experiment

```
fits <- all_growthmodels(  
  value ~ grow_userdefined(time, parms) | conc + repl,  
  data = antibiotic, p = parms, ncores = 4)  
head(results(fits))
```

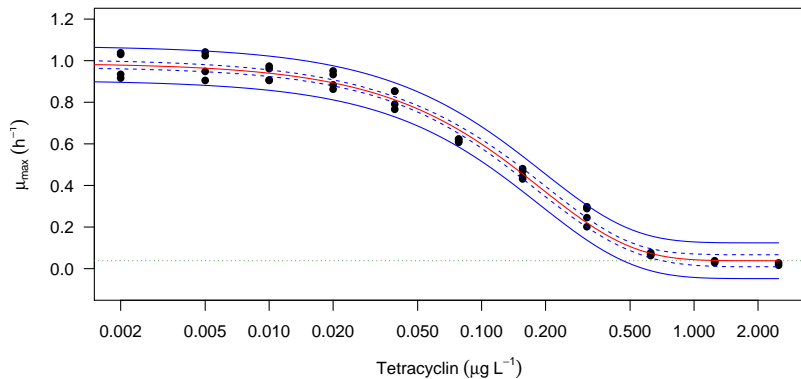
	conc	repl	y0	mumax	K	dK	r2
0:R3	0.000	R3	0.0033510	0.9616455	0.3197630	0.0070695	0.9963226
0.002:R3	0.002	R3	0.0026947	1.0302559	0.3281906	0.0068864	0.9960925
0.005:R3	0.005	R3	0.0025441	1.0416826	0.3235049	0.0070012	0.9968601
0.01:R3	0.010	R3	0.0031749	0.9735880	0.3311022	0.0069075	0.9961030
0.02:R3	0.020	R3	0.0035295	0.9330783	0.3389016	0.0068939	0.9953515
0.039:R3	0.039	R3	0.0035379	0.8530223	0.3455108	0.0063871	0.9961332

- ▶ options: box-constraints (lower, upper), selection of parameters (which), tuning of optimization methods, multi-core
- ▶ see also: all_easylinear, all_splines

All fits



Fit a functional response curve



```
res <- subset(results(fits), r2 > 0.9)
m <- nls(mumax ~ a*exp(b*conc) + c,
        data=res, start=c(a=1, b=-0.1, c=0.05))
```

- ▶ standard plotting functions
- ▶ confidence intervals with package **propagate** (Spiess, 2014)

Summary

Package growthrates

- ▶ three methods with unique interface
- ▶ plain data structure
- ▶ single and multiple fits

Convenient and Extensible

- ▶ `fit_...`, `all_...` `summary`, `plot`, `...`, `results`
- ▶ pre- and user-defined growth models
- ▶ numerically solved ODE models

→ *building blocks* for complex statistical and mechanistic models

More reading

- ▶ <https://github.com/tpetzoldt/growthrates>
- ▶ <https://cran.r-project.org/package=growthrates>

But, why not a nonlinear mixed effects model?

Nonlinear mixed effects models with **lme4** can fit both, growth rate and functional response in *one step*, given we have a good overall model, reasonable start parameters for it, and an in-depth understanding of both the methodology the data set.

Package **growthrates** can help to explore data and models and prepare us for this next step . . .

Outlook: A nonlinear mixed model with lme4

```
library("lme4")

f_custom <- function(time, conc, y0, K, dK, a, b)
  ((K + dK*time) * y0) / (y0 + ((K + dK*time) - y0) *
    exp((-a*exp(-b*conc) + 0.01) * time))

## analytical derivative
d_custom <- deriv(
  body(f_custom),
  namevec = c("y0", "K", "dK", "a", "b"),
  function.arg = f_custom
)

mm <- nlmer(value ~ d_custom(time, conc, y0, K, dK, a, b)
  ~ (K|conc:repl) + (dK|conc:repl) + (y0|conc:repl),
  data = antibiotic,
  start = c(y0=0.02, K=0.3, dK=0.005, a=0.8, b=3),
  nlmerControl(optimizer="bobyqa"))

summary(mm)
```

lme4 works :)

