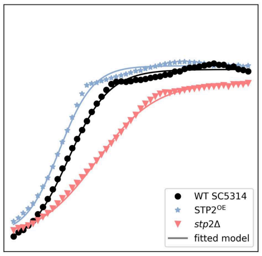
Feb 06, 2020 Version 2

Bioflux Analyses: Modelling V.2

In 1 collection

DOI
dx.doi.org/10.17504/protocols.io.bb7sirne



Tobias Weise[1], Bettina Boettcher[2], Slavena Vylkova[2]

[1]BioControl Jena GmbH, Jena, Germany;
[2]Septomics Research Center, Friedrich Schiller University and Leibniz Institute for Natural Product Research and Infection Biology – Hans Knöll Institute, Jena, Germany

Tobias Weise
BioControl Jena GmbH

DOI: dx.doi.org/10.17504/protocols.io.bb7sirne

Protocol Citation: Tobias Weise, Bettina Boettcher, Slavena Vylkova 2020. Bioflux Analyses: Modelling. **protocols.io** https://dx.doi.org/10.17504/protocols.io.bb7sirne

Manuscript citation:
This work was supported by the German Ministry for Education and Science in the program Unternehmen Region (BMBF 03Z2JN11).

Protocol status: Working
We use this protocol and it's working

Created: February 06, 2020

Last Modified: February 06, 2020

Protocol Integer ID: 32722

Keywords: Candida Albicans, Biofilm, Biofilm Formation, Growth Rate, ODE Model, Parameter Estimation

Abstract

Biofilm formation under shear flow conditions was monitored using the Bioflux1000 device (Fluxion Biosciences, Inc.). In short, Candida albicans overnight cultures were washed in pre-warmed RPMI medium. Cells were seeded for 2-5 sec from the outlet well into the channels of Bioflux1000 flow chambers, which were primed before with warm medium. The cells were allowed to adhere to the channels for 90 min without any flow, followed by removal of non-adherent cells by flowing fresh, pre-warmed RPMI medium for 5 sec. Shear flow was set for time series experiments over 24 h biofilm formation and images were captured every 20 min. Two channels were investigated in parallel having a 10 × magnification to allow a direct comparison between a mutant and a reference (wild-type) strain. Image capturing and stacks to movies was performed using the MetaMorph® Software (Molecular Devices).

An ODE model reflecting the logistic growth as well as the lag phase was fitted to the individual experiments. Fitting was carried out by minimising a cost function (unweighted least-squares-based) using the *Nelder-Mead* algorithm. Growth rate time series generated from the fitted model were used to compare wild type and mutant regarding the maximum observed growth rates at their respective time points.

All computations were performed using the programming language python (version 3.6.9) and the additional packages numpy (version 1.16.2), pandas (version 0.25.0), and scipy (version 1.3.1).

Modelling Approach

1 The modelling presented here uses a simple ODE model fitted to the individual Bioflux I_{pix} time series data sets (Fig. 1-A) in order to estimate the height as well as the time point of maximum observable growth (Fig. 2-B).

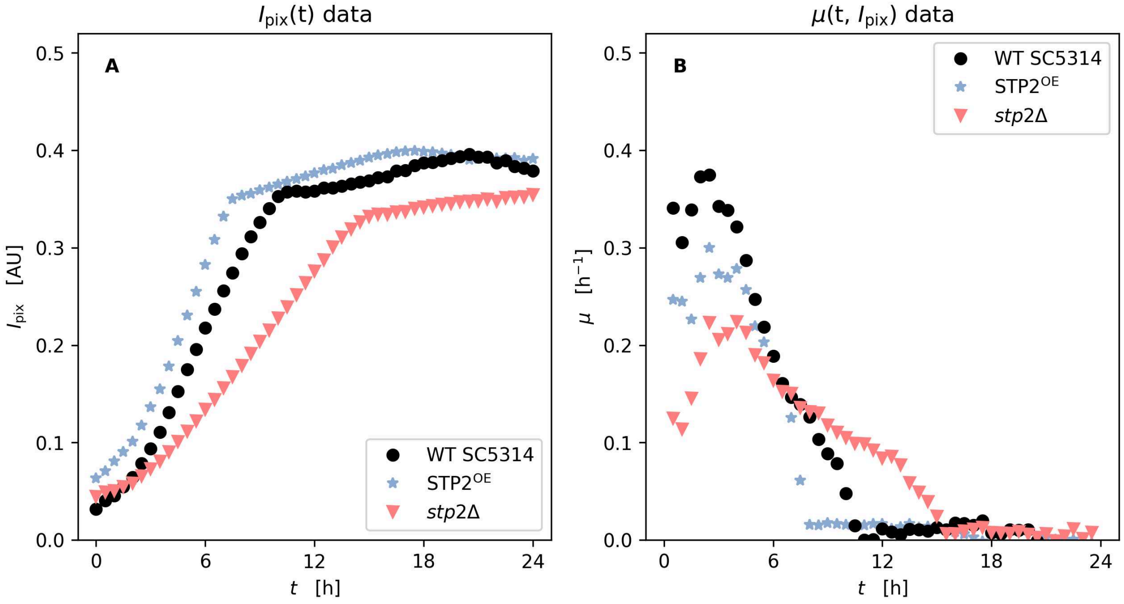
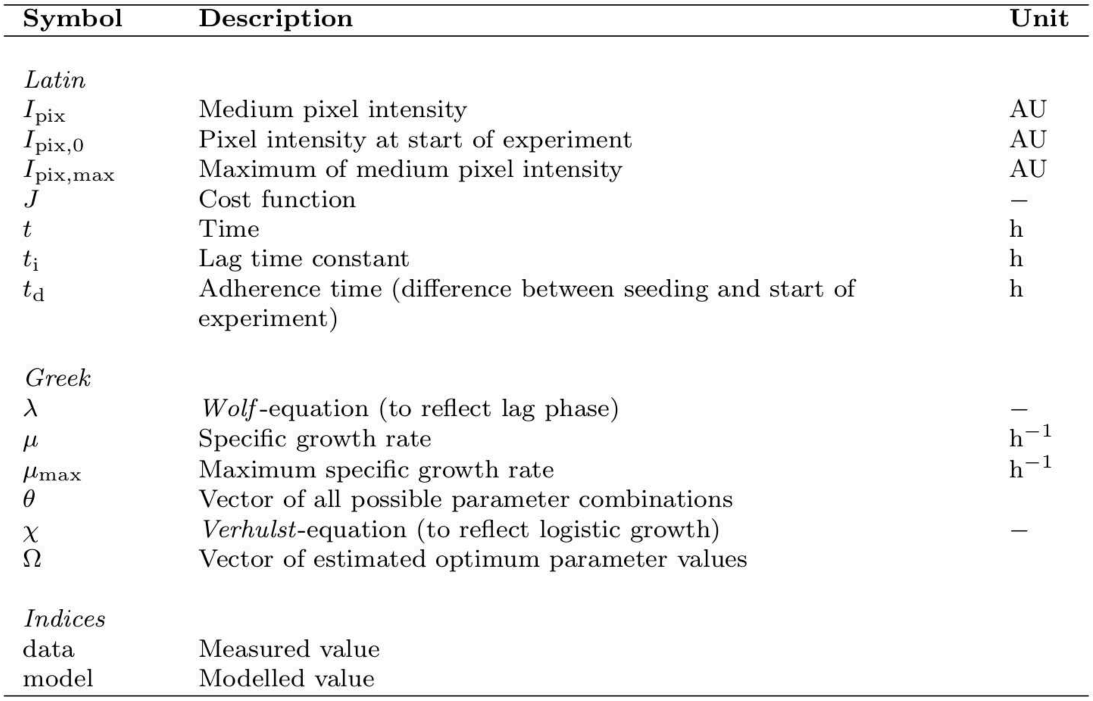


Fig. 1. Example of Bioflux time series data sets; A: medium pixel intensity; B: calculated growth rate.

Table 1 contains symbols and abbreviations used within the modelling approach and parameter estimation.



Symbol	Description	Unit
Latin		
I_{pix}	Medium pixel intensity	AU
$I_{\mathrm{pix},0}$	Pixel intensity at start of experiment	AU
$I_{\mathrm{pix,max}}$	Maximum of medium pixel intensity	AU
J	Cost function	–
t	Time	h
t_{i}	Lag time constant	h
t_{d}	Adherence time (difference between seeding and start of experiment)	h
Greek		
λ	*Wolf*-equation (to reflect lag phase)	–
μ	Specific growth rate	h^{-1}
μ_{max}	Maximum specific growth rate	h^{-1}
θ	Vector of all possible parameter combinations	
χ	*Verhulst*-equation (to reflect logistic growth)	–
Ω	Vector of estimated optimum parameter values	
Indices		
data	Measured value	
model	Modelled value	

Table 1. Symbols and abbreviations used within the modelling approach and parameter estimation.

1.1 **Ordinary differential equation for discontinuous bioprocesses**

The modelling presented here was carried out using the ordinary differential equation for discontinuous bioprocesses with respect to the mean pixel intensity calculated according to the protocol 'Bioflux Analyses: Image Preprocessing'.

$$\frac{dI_{\mathrm{pix}}}{dt} = \mu_{\mathrm{max}} \cdot \lambda(t) \cdot \chi(I_{\mathrm{pix}}) \cdot I_{\mathrm{pix}}$$

Growth Kinetics

The growth of the culture was described using the following kinetics:

- logistic equation (*Verhulst*-eqation; employed to modell the capacitive limit of the biomass accumulation)

$$\chi(I_{\mathrm{pix}}) = 1 - \frac{I_{\mathrm{pix}}}{I_{\mathrm{pix,max}}}$$

- lag phase (*Wolf*-equation; employed to modell the initial lag phase of the culture due to adaptation)

$$\lambda(t) = 1 - \exp\left(-\frac{t+t_d}{t_i+t_d}\right)$$

The model parameters μ_{max}, $I_{pix,max}$, t_i as well as the start condition $I_{pix,0}$ is estimated within Step No. 2. The adherence time t_d represents the time difference between the seeding of the cells an the start of the experiment/simulation. t_d was set to 1.5 h for all simulations in accordance to the experimental set-up.

1.2 Import of the required packages.

```
import numpy as np
import pandas as pd
import scipy.integrate
from scipy.optimize import fmin
```

1.3 Definition of the growth kinetics.

```
# logistic equation (Verhulst)
def chi(Ipixmax,Ipix):
    return 1 - Ipix / Ipixmax

# description of lag phase (Wolf-term)
def wolf(ti,td,t):
    return 1 - np.exp( - ( t + td ) / ( ti + td ) )

# growth kinetics
def mu(mumax,Ipixmax,ti,td,Ipix,t):
    return mumax * chi(Ipixmax,Ipix) * wolf(ti,td,t)
```

1.4 Definition of the ODE model.

```
# Model
#=================================================================
==========
def eq(indep_par,exp_par,start_t,end_t,incr):

    initial_cond = indep_par[0],0
    mumax,Ipixmax,ti = indep_par[1:6]
    td = exp_par

    t  = np.linspace(start_t, end_t,incr)   # time grid
#   --------------------------------------------------------------
----------
    def funct(y,t):                         # ODE system
        Ipix = y[0]                         # biomass
        tin = y[1]                          # internal t (needed
for Wolf-term)

        dIpixdt = mu(mumax,Ipixmax,ti,td,Ipix,tin) * Ipix
        dtindt = 1

        return [dIpixdt,dtindt]
#   --------------------------------------------------------------
----------
    ds = scipy.integrate.odeint(funct,initial_cond,t) #integrate
    return (ds[:,0],t)
```

Parameter Estimation

2 Based on the ordinary differential equation above, the experimental I_{pix} time series data sets were simulated individually for the respective experiment. The time series were simulated for an experiment time of 24 h. Numerical integration was carried out employing lsoda from ODEPACK (scipy.integrate.odeint).

The objective of the parameter estimation was to identify the parameter values for the respective experiment. The model parameters μ_{max}, $I_{\text{pix,max}}$, t_{i} as well as the start condition $I_{\text{pix,0}}$ are vectorised as θ. The optimum parameter values Ω were estimated by minimising the cost function J(θ) using the downhill-simplex method (Nelder-Mead method; scipy.optimize.fmin). Standard settings regarding the convergence criteria were used with respect to the above package.

$$J(\theta) = \sum_{i=1}^{n} \left(I_{\text{pix,data}} - I_{\text{pix,model}}\right)^2$$

$$J(\Omega) = \min_{\theta} J(\theta)$$

2.1 Definition of the scoring function (least-squares).

```
# definition of scoring function (least-squares beased)
def score(parms):

    # score difference between model and data points
    ss = lambda data,model:(( data - model )**2 ).sum()

    # get solution to system
    Ipix_model,t_model = eq(parms,exp_rates,0,time,100)

    # model index to compare to data
    mt=np.linspace(0,time,100)
    findindex=lambda x:np.where(mt>=x)[0][0]
    Ipix_index=list(map(findindex,data.time))
    data.Ipix_model = Ipix_model[Ipix_index]

    return ss(data.Ipix,data.Ipix_model)
```

2.2 Execution of the parameter estimation.

```
initials = [Ipix_zero,mumax,ti,Ipix_max]
exp_rates = 1.5 # [h]
time = 24 # [h]

answ = scipy.optimize.fmin(score, rates, full_output=1, retall=1,
maxiter=1000)

estimates = answ[0]
score_value = answ[1]
```

3 Based on the parametrised model Ipix and growth rate time series were simulated for each experiment (Fig. 2-A and B). The subsequent evaluations were carried out using the modelled time series.

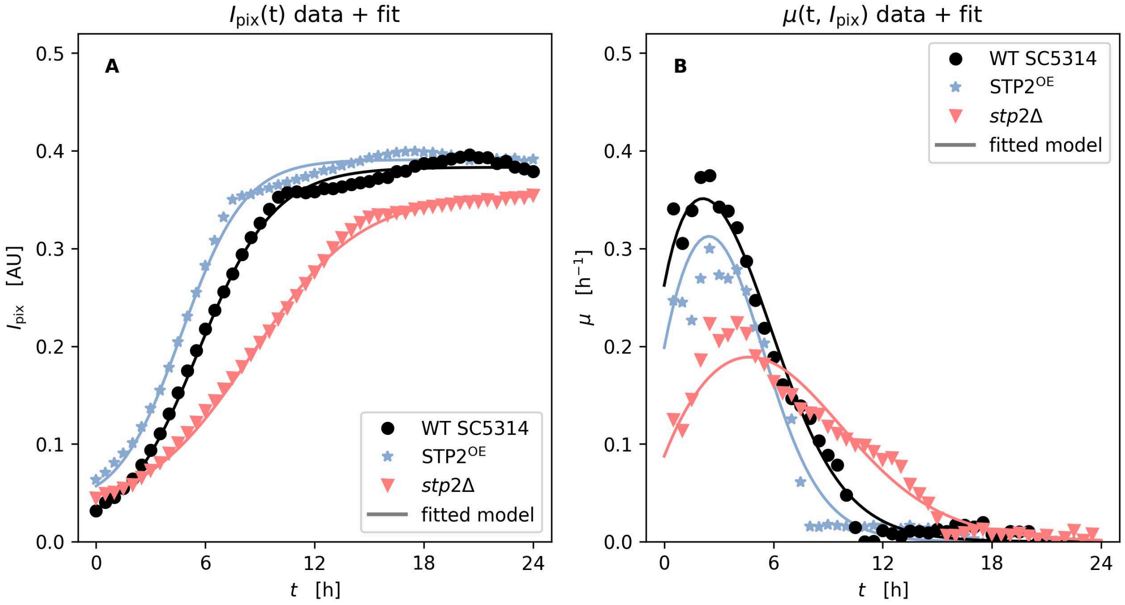


Fig. 2. Example of the fitted Bioflux time series data sets; A: medium pixel intensity; B: growth rate.

According to Fig. 2-B, the height as well as the time point of the maximum observable growth rate is recorded. These values are used for the comparison between the wild type an their respective mutant.

Case Study

4 In order to illustrate the differences in the biofilm formation between wild type and mutant strains, the observed maximum growth rates are plotted against their respective time points (Fig. 3.1)-A. The differential plot (Fi. 3.1-B), conferring to subfigure A, presents these differences between the mutant strains (arrow head) and their respective wild types (arrow tail).

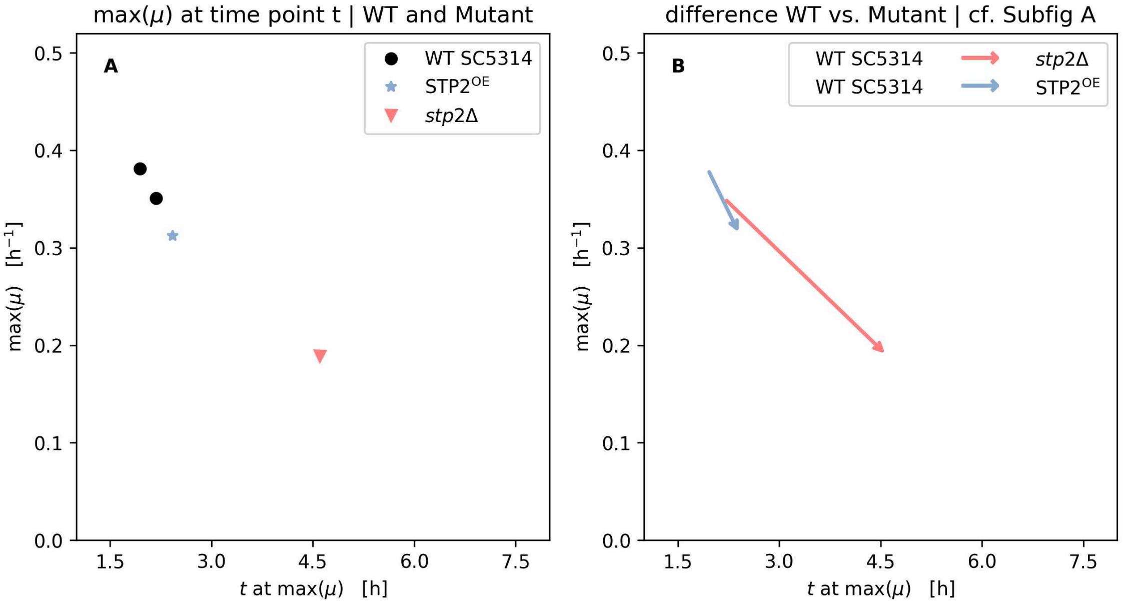


Fig. 3.1 Maximum observable growth rates max(μ) at their respective time points for wild type and mutant strains; A: Scatter plot of max(μ) at time point t from two examples (wild type vs. mutant); B: Difference plot for max(μ) at time point t, wild type (arrow tail) to respective mutant (arrow head).