Appendix B

Growth Curve Analysis

A Python script was written to automate analysis of growth curves, and ipython notebooks were developed to plot the data and analysis results from multiple experiments. See Section 4.4 for details on how data are collected and calibrated. The files described in this appendix are available at https://github.com/nwespe/OD_growth_finder/.

B.1 Population growth rates from OD curves

This program was written to extract growth rate parameters from optical density time series readings of samples in a 96-well or 384-well plate. The program determines maximum growth rate, lag time and saturation time, as well as an "effective" growth rate for any specified time period. The program can be run from the command line or an ipython notebook; examples of running the program are shown in notebook format. Formatting of input files, program options, and examples of output are described below.

The raw data file contains timepoint information in the first column and optical density values in each subsequent column. The first row contains headings with well labels. Timepoints can be formatted as HH:MM:SS or as a number indicating minutes elapsed (e.g., 0, 10, 20). The input file can be an Excel (.xlsx), CSV or tab-delimited text file.

	A	B	C	D	E	F	G
1	Kinetic read	A1	A2	A3	A4	A5	A6
2	0:00:00	0.157	0.156	0.158	0.159	0.16	0.147
3	0:10:00	0.163	0.159	0.163	0.164	0.163	0.15
4	0:20:00	0.169	0.165	0.17	0.172	0.168	0.153
5	0:30:00	0.176	0.172	0.18	0.179	0.173	0.157
6	0:40:00	0.183	0.179	0.187	0.189	0.177	0.162
7	0:50:00	0.189	0.187	0.197	0.198	0.183	0.166
8	1:00:00	0.197	0.193	0.206	0.205	0.186	0.17
9	1:10:00	0.207	0.203	0.216	0.219	0.192	0.173
10	1:20:00	0.218	0.213	0.23	0.233	0.196	0.183

An optional plate layout Excel file can contain any metadata describing the well contents; these will be associated with each sample in the results output. The first column contains the well labels, and subsequent columns contain sample information.

	A	B	C	D	E	F
1	well	strain	media	replicate	expt_date	run
2	A1	006	YPD	1	10/5/16	YPD+/-NaCl
3	A2	006	YPD	2	10/5/16	YPD+/-NaCl
4	A3	003	YPD	1	10/5/16	YPD+/-NaCl
5	A4	003	YPD	2	10/5/16	YPD+/-NaCl
6	A5	035	YPD	1	10/5/16	YPD+/-NaCl
7	A6	035	YPD	2	10/5/16	YPD+/-NaCl

Blank values are essential for accurately calculating growth rate. These can be input in several ways: as a single value; as one or more wells, in which case the average value over all timepoints is used as the blank value; or as an Excel file containing a single value for each well. The last option enables the use of a different blank value for different wells, as in the case of using multiple media types with different background optical densities in the same experiment.

	A	B	C	D	E	F
1	A1	A2	A3	A4	A5	A6
2	0.098	0.098	0.098	1.020	1.020	1.020

Here is how to run the main analysis in an ipython/Jupyter notebook. The analyze_experiment function creates an output Excel file, a summary text file, and, if sample_plots = True, an image file for each sample with the fit parameters plotted onto the raw data. The make_plots function generates a histogram and a heatmap displaying a summary of the entire experiment.

```
import growth_curve_analysis as od
```

```
expt = od.analyze_experiment('/Users/nwespe/Desktop/GROC/ypd nacl data.txt',
                             plate_layout = '/Users/nwespe/Desktop/GROC/ypd nacl layout.xlsx',
                             blank_file = '/Users/nwespe/Desktop/GROC/gen_blank_file.xlsx',
                             out_dir = '/Users/nwespe/Desktop/GROC/test/',
                             sample_plots = True, correction = [0.6, 0.2141, 1.7935])
```

```
initialized experiment
analyzed samples
created output data table
```

```
od.make_plots(expt)
```

Below is an example of running the "effective growth rate" method, with input options for start and end times in minutes.

```
expt2 = od.analyze_experiment('/Users/nwespe/Desktop/GROC/ypd nacl data.txt',
                             plate_layout = '/Users/nwespe/Desktop/GROC/ypd nacl layout.xlsx',
                             blank_file = '/Users/nwespe/Desktop/GROC/gen_blank_file.xlsx',
                             out_dir = '/Users/nwespe/Desktop/GROC/',
                             method = 'effective_growth_rate',
                             start = 120, end = 360, saturation = True,
                             sample_plots = True)
od.make_plots(expt2, save = True,
              metric1 = 'growth rate 120-360', unit1 = 'ln(2)/minutes',
              metric2 = 'growth rate 120-360', unit2 = 'ln(2)/minutes')
```

```
initialized experiment
analyzed samples
created output data table
```

The output Excel file lists the calculated growth rate and details of this calculation for each sample.

	A	B	C	D	E	F	G	H
		well	growth rate	r-squared	doubling time	time of max growth rate	start of fit region	end of fit region
49		E1	0.00638313	0.99777214	108.590499	110	4	19
50		E2	0.00631378	0.99691896	109.783178	110	3	19
51		E3	0.00511805	0.99858381	135.431847	130	5	21
52		E4	0.00549975	0.99750496	126.032507	130	5	21
53		E5	0.00269513	0.99607891	257.185323	150	6	25
54		E6	0.00269228	0.99793209	257.457187	170	6	29

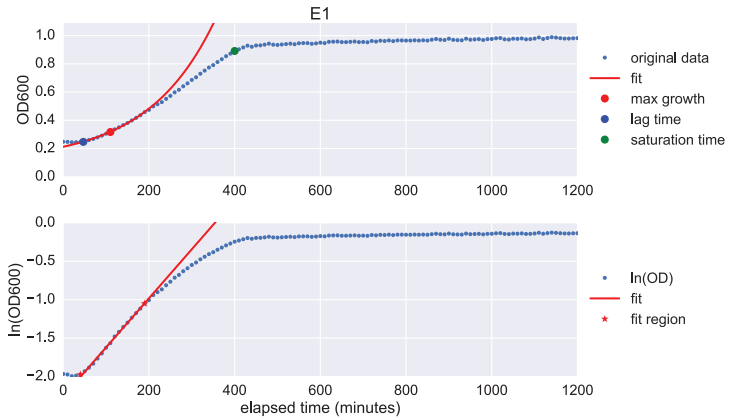
The output file also includes lag time, saturation time, and maximum OD reached.

	A	B	I	J	K	L	M	N
		well	lag time	OD at end of lag	saturation time	OD at saturation	max OD	time of max OD
49		E1	46.8515814	0.247	400	0.783	0.989	1140
50		E2	37.6644629	0.273	390	0.834	1.016	1140
51		E3	52.7314242	0.268	470	0.742	0.891	1180
52		E4	54.6872237	0.241	490	0.726	0.882	1140
53		E5	65.9894846	0.222	940	0.425	0.545	1140
54		E6	64.2155202	0.253	730	0.487	0.621	1140

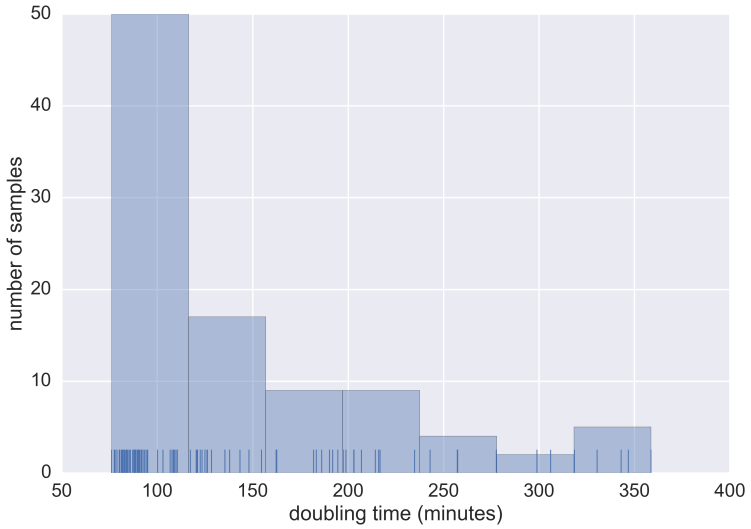
The last columns in the output file include information from the optional plate layout file.

	A	B	O	P	Q	R	S	T	U
		well	row	column	name	media	replicate	expt_date	run
49		E1	5	1	003	0.5 M NaCl	1	Jul 27 2016	YPD+/-NaCl
50		E2	5	2	003	0.5 M NaCl	2	Jul 27 2016	YPD+/-NaCl
51		E3	5	3	006A	0.5 M NaCl	1	Jul 27 2016	YPD+/-NaCl
52		E4	5	4	006A	0.5 M NaCl	2	Jul 27 2016	YPD+/-NaCl
53		E5	5	5	110-c4	0.5 M NaCl	1	Jul 27 2016	YPD+/-NaCl
54		E6	5	6	110-c4	0.5 M NaCl	2	Jul 27 2016	YPD+/-NaCl

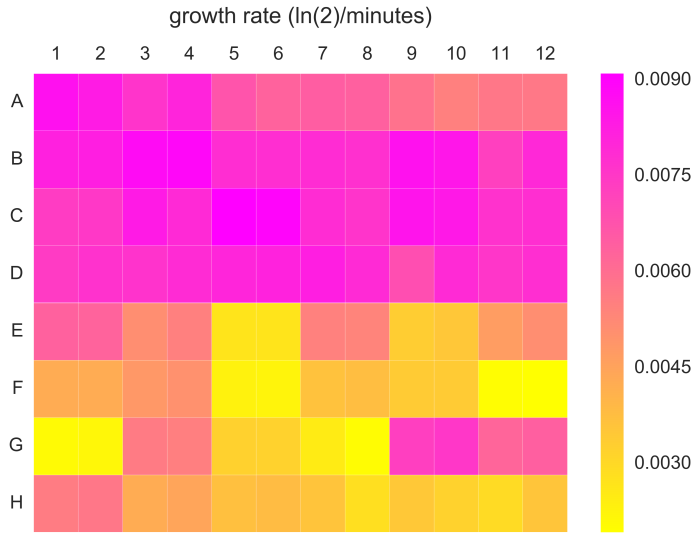
If sample_plots = True, an SVG file is created for each sample showing the calculated parameters plotted with the raw data.



Any numeric column of the output file can be plotted in a histogram or heatmap using the function "make_plots." The histogram shows the distribution of values of a given output parameter for the experiment. The default parameter is doubling time. Other parameter options include growth rate, lag time, saturation time, and max OD.



The heatmap shows values of a given output parameter for each well in the experiment. The default parameter is growth rate. Here, the usage of duplicate wells for each strain can be clearly seen, especially in rows E through H.



Growth rates: input options

Parameter	Description	Default value
data_file	file containing OD data	None (required)
plate_layout	file containing sample metadata	None
blank	blank value	0
blank_file	file containing blank value by well; overrides blank	None
method	see descriptions in separate table	'sliding_window'
out_dir	destination for output files	current directory './'
window_size	number of OD values used in sliding window to fit linear regression	9
sample_plots	create sample plots (adds significantly to runtime)	False
droplow	drop very low values from analysis (below -4.6 after calibration, equal to OD 0.01)	False
start	start time for "effective growth rate"	0
end	end time for "effective growth rate"	None (uses last point)
saturation	use saturation point as end time for "effective growth rate" if sample saturates before specified end (recommended)	False
correction	parameters for correcting non-linearity of OD readings; input as list [A, B, C] where A is OD value above which correction will be applied, and B and C are from the exponential fit y = B * exp(C*x) of measured vs. expected OD values, e.g., [0.6, 0.2141, 1.7935]	None

Growth rates: calculation method options

Method	Description
'sliding_window' (default)	Finds maximum growth rate. First uses a sliding window to find the maximum slope of the log-transformed data. Calculates maximum growth rate as slope of linear regression fit to all points whose sliding-window slopes were within 90% of the maximum.
'smooth_n_slide'	Fits a spline to all data points, then finds maximum growth rate using a sliding window as above. Used internally by 'sliding_window' method for samples whose r-squared value is below 0.9 in initial calculation.
'spline'	Fits a spline to all data points; maximum growth rate is the maximum derivative of spline. Faster but less accurate than sliding window methods.
'effective_growth_rate'	Calculates growth rate by fitting an exponential curve to all data points within specified start and end times.

B.2 Plotting growth rates results

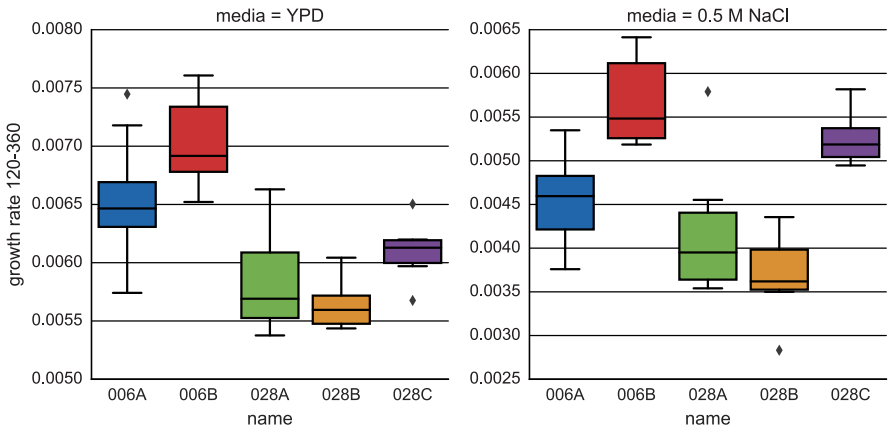
An ipython notebook was developed to generate boxplots summarizing the growth rate results from multiple experiments. The input files for this notebook are the results Excel files created by the program growth_curve_analysis.py described in Section B.1. The key steps executed by this notebook are described below.

All results to be plotted are first imported and compiled into a single dataframe. User specifies a subset of the data to be plotted using a list of identifiers.

```
strain_list = ['006A', '006B', '028A', '028B', '028C']
group = 'all_ancestors'
```

The results for a given metric are grouped by identifier, segregated by condition, and plotted onto a boxplot.

```
g = sns.factorplot(x='name', y='growth rate 120-360',
                   col='media', col_order=['YPD', '0.5 M NaCl'],
                   data=gc, order=strain_list,
                   kind='box', legend=False, sharey=False)
plt.savefig('/Users/nwespe/Desktop/'+group+'_boxplot.svg', format='svg', bbox_inches='tight')
```



The metric (y) can be any numeric column of the results file (e.g., saturation time, lag time). The identifier (x) and condition (col) are descriptors from the plate layout file.

B.3 Aggregate growth curve plots

An ipython notebook was developed to create aggregate curves using the seaborn (version 0.7.1) data visualization package for python (seaborn.pydata.org). The input files for this notebook are the same as for growth_curve_analysis.py. See seaborn.tsplot for more details. The key steps executed by this notebook are described below.

The raw data and plate layout information are imported for each experiment.

To plot data from multiple wells and experiments together, the data are normalized by dividing each OD value by the initial OD. Dataframes are created of the normalized data and the log-transformed normalized data.

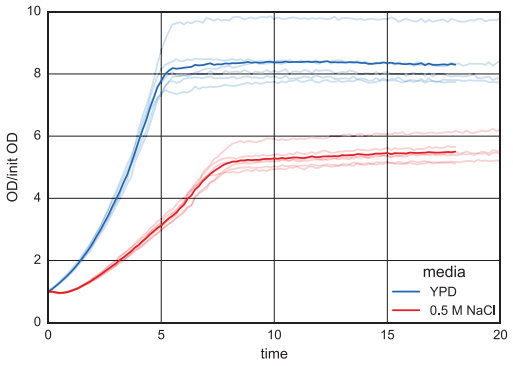
User specifies a subset of the data to be plotted using either a single identifier or a list of identifiers, e.g., strain names. The first example is a plot for a single strain with data separated by media type.

```
strain = '006A'
```

```
subset = trimmed_norm_data[trimmed_norm_data.name == strain]
lf_norm_data = get_long_form(subset, 'OD/init OD')

fig = plt.figure()
ax = sns.tsplot(data=lf_norm_data, time='time', value='OD/init OD', unit='sample',
                condition='media', err_style='unit_traces')
```

The data are plotted with a light line for each replicate and a dark line representing the mean value of the replicates at each timepoint with interpolation between the timepoints.



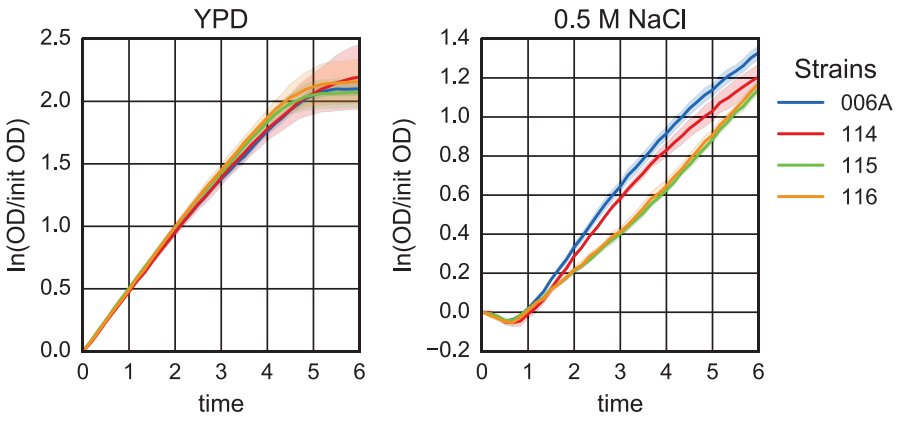
The following examples are plots of data for multiple strains. A time subset of the data can also be used; e.g., the first 6 hours of an 18-hour timecourse.

```
strain_list = ['006A', '114', '115', '116']
group = 'non-mutator evolved'
```

```
data = trimmed_log_norm_data  # or trimmed_norm_data
val = 'ln(OD/init OD)'  # or 'OD/init OD'
```

```
# to plot a time subset of the data
first_three_hrs = pd.concat([data.iloc[:, 0:19], data.iloc[:, -4:]], axis=1)
first_six_hrs = pd.concat([data.iloc[:, 0:37], data.iloc[:, -4:]], axis=1)
first_twelve_hrs = pd.concat([data.iloc[:, 0:73], data.iloc[:, -4:]], axis=1)

data = first_six_hrs
```



In the figure above, data for different media are separated onto two plots. These can also be plotted onto a single figure as shown below. Shaded regions around the lines show 95% confidence intervals.