Growth rate algorithm

Danielle Carpenter

Brief description and method

This algorithm takes a .csv file that you convert from the output of the BioTek (or any) plate reader and finds the region of highest slope on a log/linear plot.

Install R and R Studio. Arrange your .csv file with time across rows and samples in columns. You may have any number of columns with labels; note the column number that contains the label you actually want on the plots, and note the column number that contains the beginning of the data. Place this file in your desired working directory. Download, unzip and import the files accompanying these instructions into the same working directory.

Load analysis.r into R. Follow comments in analysis.r according to the parameters of your .csv file and desired file names. Run the file and inspect each plot to ensure a reasonable area was chosen for slope determination. Doubling time is ln2/slope (m).

Note added 6/13: Pat Gibney observed that the calculated growth rate varies with the number of interval (int) points chosen. This is most evident for faster growing cells. We have adjusted the default int to 12 for 15 minute time points, to minimize this variation.

Detailed description and methods

This function can be called in R to analyze OD600 data from a plate reader. It requires basic proficiency with the software language R. It takes in OD600 data points and their associated time points from one well of data. It then process this data outputs the associated exponential phase slope, the coefficient of determination of the linear fit (R^2), and the length of the lag phase. It also generates a plot of the data.

Data Input

Imagine you have a vector of OD600 data points and a vector of associated time points in R.

```
od600 <- c(0.1, 0.1, 0.2, 0.25, 0.3, …)

time <- c(0, 15, 30, 45, 60, …)
```

You can run the function on this data by running the following lines of code.

```
source("Pathtofile/find_gr.R")

output <- findgr(od600, time, "Experiment 1", int=12, r2.cutoff=0.997)
```

The first line only needs to be run once in a session and loads the function so R can recognize it. The second line is the actual call of the function, and saves the output data to a variable called `output`. The arguments in order are:

- Vector of OD600s
- Vector of time
- String that will be the title of the plot of data
- the number of points you want to use to calculate log phase growth. Default is 8 (2 hours when sampling every 15 minutes), shorter is more lenient, longer is more stringent.
- The cutoff of R^2 for how good of a fit you would like. I chose 0.997 as a default, but you may lower this if the data is very noisy.

Batch Processing

The first step in batch processing is to export your data in a table format from the plate reader software. I then open the data up in Excel, and modify as needed. I generally use the following format

Well	Name	0	15	30
A1	Sample1-1	0.001	0.053	0.105
A2	Sample1-2	0.001	…	…
A3	Sample1-3	…		
A4	Sample2-1			
A5	Sample2-2			
A6	Sample2-3			
…	…			

and export my data as a CSV file. I then load the CSV file into R,

```
dat <- read.csv("myfile.csv")
```

generate a time vector,

```
time <- seq(0,360, by=15)
```

and then run a “for” loop on the data table.

```
M = nrow(dat)

N = ncol(dat)


pdf("growth_rate_plots.pdf", paper="letter", width=7.5, height=10)

for (i in 1:M) {

      gr <- findgr(dat[3:N, i], time, dat[2, i], int=12, r2=0.997)

      growth.rates <- rbind(growth.rates, gr)

}

dev.off()


write.table(growth.rates, "growth_rates.txt", sep="\t", quote=F,
row.names=F)
```

The plots will be stored in a file called “growth_rate_plots.pdf” and the rates themselves will be stored in a tab-delimited text file called “growth_rates.txt” that can be opened in Excel.

How things work

First, the line is tested to see if it is basically flat (slope < 0.0001). If so, then it is considered to have no growth. Otherwise, the OD600 data is log transformed, and for every possible interval of `int` (default 12) points, a linear regression is carried out and the slope and R^2 are stored in a matrix.

Then, each entry in the matrix of slopes which has an R^2 of less than the cutoff (default 0.997) is dropped from consideration. Finally, the entry with the maximum slope is saved and kept as the best fit.

I then search outward from the interval with maximum slope. Each point is considered part of the exponential growth phase if its residual off the line of best fit is less than 5%. The lag time is the earliest time point at which the residual is greater than 5% from the line of exponential growth.

Below is a sample plot illustrating the output of the function. Each point sampled is plotted in black on a semi-log scale. The points outlined in red represent the interval which had the maximum slope. The dashed red line is the line of best fit. The vertical light blue dashed lines represent the exponential growth phase. In the background of this plot are grey dots which represent the slope of the line at each time point. I've commented this feature out in the code I've included, so you don't need to worry about it.

If you have any questions, please feel free to contact me!