AJB2021_FVT_CEPF_S2

Robert L. Baker

12/29/2020

Abstract

This document includes links to an example dataset and code for employing an integral approach to function valued trait data.

Background and explanation

The data used here are from Li et al, 2020. “Epistatic Transcription Factor Networks Differentially Modulate Arabidopsis Growth and Defense” Genetics vol. 214 no. 2 529-541; https://doi.org/10.1534/genetics.119.302996 (https://doi.org/10.1534/genetics.119.302996).

The authors have time-series data on plant growth. When they model these data they find that the growth of different genotypes is best fit by different functions. Therefore, the FVT approach taken where a function is fit to data and parameters of that function are used “as data” won’t work because the functions are different."

Rather than using the parameters of an individual function “as data” we find the area under the curve described by each function and use that area “as data”.

We run a couple of quick correlations test to see whether area under the curve is correlated with early as well as late time points. Not unsurprisingly, area under the curve is very strongly correlated with late developmental time points. However, diff is also significantly and strongly correlated with early developmental time points.

It should be noted that the data used here are LS means for each day rather than replicate level data. Depending on the goal of the analysis, it is equally possible to find the area under the curve for individual replicates and then calculate LS means for the area under the curve.

```
rm(list=ls()) #remove all objects from the R environment
gc() #can help return memory to R after large objects have been called.
```

```
library(tidyverse)
library(readxl)
library(filesstrings)
```

Download the files, unzip & remove whitespace from filenames:

```
#set working directory & load data:
#download zipped supplimental files at: https://gsajournals.figshare.com/ndownloader/files/20239863
#unzip files and set working directory to the folder "File S1" (or wherever you have saved the unzipped files)

remove_filename_spaces(replacement = "_") #replaces all whitespace in filenames with "_"
```

Load files and generate a new working dataframe.

The excel files contains 2 rectangles of data from separate growth chambers and these will need to be loaded separately.

```
growth_CEF<-read_excel("Supplemental_Data_Set_1,_lsmeans_of_growth,_flowering_and_GLS
_for_single_mutants.xlsx", range="B6:N27")
growth_CEF<-cbind("CEF", growth_CEF[,c(1,3:length(growth_CEF))])
colnames(growth_CEF)[1]<-"chamber"

growth_LSA<-read_excel("Supplemental_Data_Set_1,_lsmeans_of_growth,_flowering_and_GLS
_for_single_mutants.xlsx", range="B32:N53")
growth_LSA<-cbind("LSA", growth_LSA[,c(1,3:length(growth_LSA))])
colnames(growth_LSA)[1]<-"chamber"

#generate a single dataframe with data from both growth chambers.
growth<-rbind(growth_CEF, growth_LSA)

#pull out growth data and store it as a list.
growth.list<-as.list(as.data.frame(t(growth[,4:13])))
```

Test case with graphical interpretation.

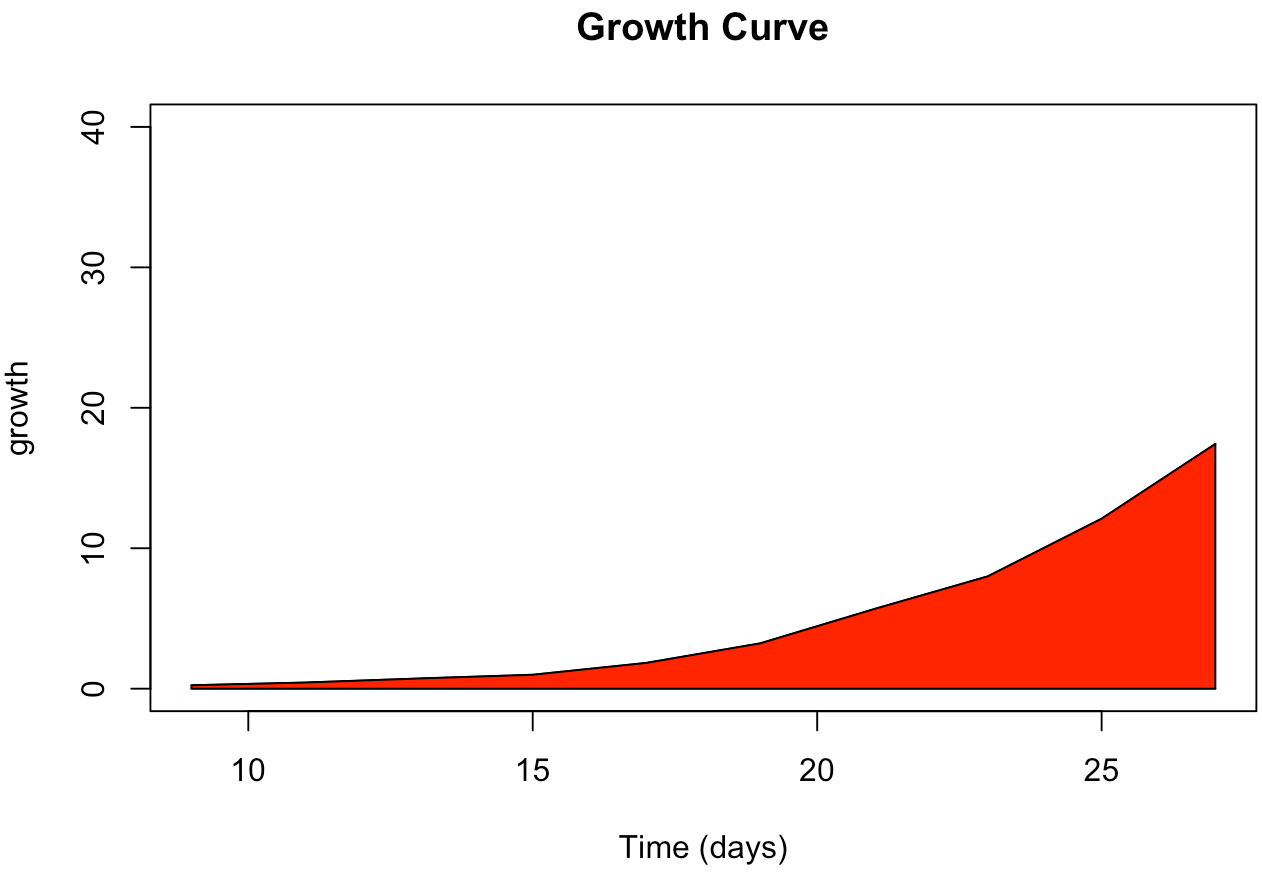
Here, we use just the first line of data to calculate the area under the curve, which we designate “diff”.

```
y1<-growth.list$V1 #grab growth data from the first line of the data set
y2<-rep(0, length(y1)) #set a baseline, in this case zero (but you could also compare
two functions)
x<-c(9,11,13,15,17,19,21,23,25,27) #x-axis time values that correspond to y-axis grow
th data; hard coded for simplicity

f1<-approxfun(x,y1-y2) #approximate the function of the relationship between growth a
nd time
f2<-function(z)abs(f1(z)) #absolute value of the difference between the baseline valu
e and the growth data in question. The absolute value doesn't make much difference wi
th baseline of zero but might be important if you wanted to directly compare 2 genoty
pes whose growth function intersected at some point - or a single genotype in multipl
e environments with different growth functions that intersected.

plot(x,y1,type="l",ylim=c(0,40), ylab="growth", xlab="Time (days)", main="Growth Curv
e") #plot the estimated function

lines(x,y2,type="l",col="red") #plot the "baseline" value - zero in this case
polygon(c(x,rev(x)),c(y1,rev(y2)), col="red") #fit a polygon to the area between the
function and the baseline: in this case the polygon represents the growth that the fi
rst individual underwent over time.
```




```
diff<-integrate(f2,min(x),max(x)) #do some calculus to find the area of that polygon

diff
```

```
## 83.63555 with absolute error < 0.0038
```

Do this for the entire dataset (but skip the plots):

```
diff<-NULL #set up a couple of empty variables
err<-NULL #set up a couple of empty variables
for(i in 1:nrow(growth)){
    y1<-get(paste("V", i, sep=""), growth.list)
    y2<-rep(0, length(y1))
    x<-c(9,11,13,15,17,19,21,23,25,27)

    f1<-approxfun(x,y1-y2)
    f2<-function(z)abs(f1(z))
    dif<-integrate(f2,min(x), max(x))

    diff[length(diff)+1]<-dif$value #append the current differential to "diff"
    err[length(err)+1]<-dif$abs.error   #append the current error estimate to "err"
    }

#add the two new variables, diff and err to the datset
dat1<-cbind(growth, diff, err)
```

Run a couple very simple correlation tests to compare the “area under the curve” (diff) to individual values used to generate the curve (first and last values):

```
cor.test(dat1$day27, dat1$diff) # test the correlation between the area under the fun
ction-value-trait curve and the last day of data collection.
```

```
##
##  Pearson's product-moment correlation
##
## data:  dat1$day27 and dat1$diff
## t = 42.725, df = 40, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.9799014 0.9942311
## sample estimates:
##       cor
## 0.9892204
```

```
#Diff is correlated to that final value. What about the first value? We would expect
that contributes relatively little to the area under the curve, based on the figure a
bove.

cor.test(dat1$day9, dat1$diff)
```

```
##
##  Pearson's product-moment correlation
##
## data:  dat1$day9 and dat1$diff
## t = 10.762, df = 40, p-value = 2.226e-13
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.7564283 0.9239693
## sample estimates:
##       cor
## 0.8621495
```

```
#The correlation is, as expected, not as strong. But an r of 0.762 is still pretty go
od, and it is highly significant.
```

How well does the “area under the curve” (integral) apporach correlate with the slopes calculated in Li et al 2020?

```
cor.test(dat1$slope, dat1$diff)
```

```
##
##  Pearson's product-moment correlation
##
## data:  dat1$slope and dat1$diff
## t = 7.9334, df = 40, p-value = 9.735e-10
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.6270123 0.8773563
## sample estimates:
##       cor
## 0.7819319
```

```
#The correlation is not as strong as with individual datapoints, but is still substan
tial and certainly significant.
```

Citations and documentation

```
##
##   R Core Team (2020). R: A language and environment for statistical
##   computing. R Foundation for Statistical Computing, Vienna, Austria.
##   URL https://www.R-project.org/.
```

```
## [1] "R-4.0.2_2020-06-22"
```

```
##
##   Wickham et al., (2019). Welcome to the tidyverse. Journal of Open
##   Source Software, 4(43), 1686, https://doi.org/10.21105/joss.01686
```

```
##
## To cite filesstrings in publications use:
##
##   Rory Nolan and Sergi Padilla-Parra (2017). filesstrings: An R package
##   for file and string manipulation. The Journal of Open Source
##   Software, 2(14).  DOI: 10.21105/joss.00260.
```

```
##
## To cite package 'readxl' in publications use:
##
##   Hadley Wickham and Jennifer Bryan (2019). readxl: Read Excel Files. R
##   package version 1.3.1. https://CRAN.R-project.org/package=readxl
```