Users Guide for "implant" Package

Ronghao Wang[1], Yumou Qiu[2], Yuzhen Zhou[1], Yuhang Xu[3], and James C. Schnable[4]

[1]Department of Statistics, University of Nebraska-Lincoln, Lincoln, 68503, USA
[2]Department of Statistics, Iowa State University, Ames, 50011, USA
[3]Department of Applied Statistics and Operations Research, Bowling Green State University, Bowling Green, 43403, USA
[4]Center for Plant Science Innovation, Department of Agronomy and Horticulture, University of Nebraska-Lincoln, Lincoln, 68503, USA

ABSTRACT

This users guide provide examples for the R package *"implant"*. The main results and methodology are presented in the paper "A High-throughput Phenotyping Pipeline for Image Processing and Functional Growth Curve Analysis".

Contents

1 Introduction **2**
2 Downloading the Package **2**
3 Image Processing **2**
3.1 Loading Example Images from the Package . . . 2
3.2 Defining the Region of Interest . . . 2
3.3 Image Segmentation using Thresholding Method . . . 3
3.4 Morphological Operations . . . 3
3.5 Obtaining the Final Segmented Image . . . 4
3.6 Extracting the Phenotypical Parameters . . . 4
4 Functional ANOVA **4**
4.1 Preparing for the Model Fitting . . . 4
4.2 Model Fitting . . . 5
4.3 Statistical Inference . . . 5
5 FANOVA Example with an Interaction Term (Example 2) **6**
6 Appendix **7**

1 Introduction

This tutorial provides a guide of the *implant* package for processing RGB images of plants and using functional anova to analyze the growth dynamics for the plants. The whole procedure can be devided into two part, the data pre-processing part and functional analysis part. In the data pre-processing part, we first process the images of the plants of interest taken from the green house and using a series of steps to process these images to segment the bodies of these plants from their backgrounds. Then we can extract the phenotypical parameters such as size, height and width, etc., from the processed images. In the functional analysis part, we use the extracted features from the processed images as our data and use the functional anova approach to build models and have the corresponding analysis.

2 Downloading the Package

The package *"implant"* is uploaded on Github. Users can download the package using:

```
devtools::install_github("rwang14/implant")
library(implant)
```

Notice that users have to load the package *"devtools"* before using the above code.

3 Image Processing

In this part, we will use functions within this package to pre-process the RGB image data we have. Our goal is to segment the object of interest (body of the plant) from its background, then we extract the phenotypical parameters from the segmented images. Figure1(a) from the Appendix is an image of a plant in a green house, and Figure1(b) is an image with exactly the same condition as Figure 1(a) except that it has no plant in the pot. Instead, a green strip was inserted into the pot.

3.1 Loading Example Images from the Package

We stored two example png file of plants called *"image1.png"*,*"imageX1.png"*, and an image of empty pot called *"image_pot.png"* for the image processing part. In the Image Processing part, we first load a plant image and the pot image, and use them to illustrate how this part works.

```
library(png)
#load the image of plant
imageOriginal = readPNG(system.file("extdata", "image1.png",
                                    package = "implant", mustWork = TRUE))[ , ,1:3]
#load the image of empty pot
imageX = readPNG(system.file("extdata", "image_pot.png",
                                 package = "implant", mustWork = TRUE))[ , ,1:3]
```

3.2 Defining the Region of Interest

Using the package "*png*", we can transform these two images into arrays of pixel intensities. Let's name Figure 1(a) as *imageOriginal* and Figure 1(b) as *imageX*. Since these two images all contain RGB channels, the figures should be read as three-dimensional arrays with the last dimensions is equal to 3 or 4. This depends on whether the images you use are RGB (3 channels) or RGBA (4 channels). In our case, the images we use has 4 channels, and we only want to consider the RGB channels. Therefore, we take the first three channels as our data matrices instead of using all the four channels. Hence, for each image, we obtain an array of three same-size matrices. Then we can check the dimensions of these two arrays. For example, we'd like to find the dimensions for each matrix *imageOriginal* for each channel:

```
sizeOriginal = dim(imageOriginal)[c(1, 2)]
>2056 2454
```

We can see that in each channel, the matrix has 2056 rows and 2454 columns.

Functions *ColorG* and *ColorB* can help us identify the region of interest. Using the function *ColorG*, we can find the border-top of the pot, i.e., the bottom of the plant. See Figure 1(a).

```
tempG = ColorG(imageX, rowThreshold = 0.002)
lowerRowBound = tempG$lowb
```

In order to obtain a well segmented result using thresholding method, we identify the region of interest and change the color of the unwanted region into black (since we will segment the image in the next step and make the segmented image binary, i.e., coloring the background as black and the object of interest as white.). "*ColorB*" can help us identify the inner black bars, incorporated with the border-top of the pot we obtained by "*ColorG*", we can obtain the region of interest for the plants. See the red rectangle in Figure 1 panel (b). We then let the pixel intensities of the uninterested region be equal to 0 (i.e., change the color to black).

```
#Identify the region of interest
resultColor = ColorB(imageOriginal)
ColBound = c(resultColor$lb, resultColor$rb)
RI = matrix(1, sizeOriginal[1], sizeOriginal[2])
RI[, c(1 : ColBound[1])] = 0
RI[, c(ColBound[2] : sizeOriginal[2])] = 0
RI[c(lowerRowBound : sizeOriginal[1]), ] = 0
```

Notice that this identification strategy is for the dataset from the UNL greenhouse system only. Different systems need different strategies for locating the region of interest.

3.3 Image Segmentation using Thresholding Method

In order to make the process more efficient, we can reduce the sizes of the image of the plant and the image of the empty pot using function *downsize*, and name the reduced original image and the reduced empty pot image as *image*1*_reduced* and imageX1, respectively.

```
image1_reduced = downsize(imageOriginal, RowSample = 2, ColSample = 2)
imageX1 = downsize(imageX, RowSample = 2, ColSample = 2)
```

And we can see that the sizes of these two images are both reduced from 2056×2454 to 1028×1227.

We apply "*ColorG*" to imageX1 (Figure 1 (b)), we can change the color of the strip from green to white (the same color as the background). Then we can have a contrasted image (imageContrast) using the absolute value of the difference between *image*1*_reduced* and imageX1 (see 1 (c) in the Appendix).

```
tempG1 = ColorG(imageX1, rowThreshold = 0.002)
imageX1G = tempG1$c
imageContrast = abs(image1_reduced - imageX1G)[, , 1 : 3]
```

Function *imageBinary* can help us segment the object of interest from the background and transform the segmented image into black and white, where the object of interest is colored as white and the background is colored as black. By defining the parameters of this function, we can have the segmented images for *image*1*_reduced* and imageContrast, respectively. Then we can have a well segmented result by finding the intersection between the segmented *image*1*_reduced* and the segmented imageContrast. See Figure 1 (d) - (f) in the Appendix.

```
imageB1 = imageBinary(image1_reduced,  weight = c(-1, 2, -1), threshold1 = 30 / 255,
threshold2 = 0.02)
imageB2 = imageBinary(imageContrast,  weight = c(1, -2, 1), threshold1 = 0.7,
threshold2 = 0)
imageB = imageB1 * imageB2
```

Note: An alternative method for segmentation we provide in this package is the Hidden Markov Random Field-EM Method. For details, please check the help documentation for the function *"HMRF"*.

3.4 Morphological Operations

The next step is the morphological operations. Since there are still some unwanted noise, we would like to use the *erosion* function to eliminate those noise. However, this action may cause the loose information of the segmented plant. Therefore, the function *dilation* is needed as well. We follow the procedure: "Dilation-Erosion-Erosion-Dilation" and get the following result (see Figure 1 (g) in the Appendix).

```
imageBD = dilation(imageB, mask = matrix(1, 5, 5))
imageBDE = erosion( imageBD, mask = matrix(1, 5, 5) )
imageBDEE = erosion( imageBDE, mask = matrix(1, 3, 3) )
imageBDEED = dilation( imageBDEE, mask = matrix(1, 3, 3) )
```

3.5 Obtaining the Final Segmented Image

From the "Defining the Region of Interest" step, we have a matrix *RI* which has the same size as the original image, but the region that we are not interested in has been colored as black. Since the size of the original image has been reduced, we need to reduce the size of *RI* to the same size as *imageBDEED* we got from the previous step. Then we cut the processed image (imageBDEED) and keep the region of interest (in order to remove the noise around the plant body).

```
RI1 = downsize_matrix(RI, RowSample = 2, ColSample = 2)
RI1row = sum(rowSums(RI1) > 0)
RI1col = sum(colSums(RI1) > 0)
imageBF = matrix(imageBDEED[RI1 == 1], RI1row, RI1col)
```

imageBF is the final segmented image we want. See Figure 1(i) in the Appendix.

3.6 Extracting the Phenotypical Parameters

Using the obtained segmented image, we can extract the information we want from its matrix. Firstly, we need to know how many millimeters that each pixel represents horizontally and vertically. The information we have is:

```
Xsize (in mm): 1.54902601242065
Ysize (in mm): 1.55327594280243
```

which means each pixel from the image represents 1.549*mm* in horizontal and 1.553*mm* in vertical. Using this information, we can calcualte the size, height and weight from the pixel matrix of the segmented image.

```
#a = 2, b = 2 since we have reduced the size by picking every two pixels
#in each row and each column
extract_pheno(processed_image = imageBF,
        Xsize = 1.54902601242065, Ysize = 1.55327594280243, a = 2, b = 2)$height
extract_pheno(processed_image = imageBF,
        Xsize = 1.54902601242065, Ysize = 1.55327594280243, a = 2, b = 2)$width
extract_pheno(processed_image = imageBF,
        Xsize = 1.54902601242065, Ysize = 1.55327594280243, a = 2, b = 2)$size
```

Details can be found in the help documentation of this function. All the above R code can be found in the file *"demo_DCT.R"* from the *"data-raw"* folder of the package.

4 Functional ANOVA

4.1 Preparing for the Model Fitting

In the previous part, we have extracted the phenotypical parameters we want from the processed images. In this part, we will use these parameters as our data to build Functional ANOVA models.

The example data we use in this tutorial is reduced from the data being described in the paper. The reduced data includes the size of 9 plants with three genotypes (78, and 85 and 129) within 3 different blocks. These plants were observed from 0^{th} day to the 44^{th} day. We have attached this example csv file in this package called *data.csv*. We read the attached data and define the input terms as follows:

```
data_new = read.csv(system.file("extdata", "data.csv",package = "implant",
                    mustWork = TRUE))
#The first column is the positions of the observations from the original dataset,
#which can be ignored.
Y.na.mat = data_new[,-c(1:3)]
#This step is to factorize each factor
Genotype = as.factor(data_new$Genotype)
Block = as.factor(data_new$Block)
X = data.frame(Genotype, Block)
formula = "~ Genotype + Block"
tt = seq(from = 0, to = 44,by = 2)
```

where X is a dataframe of explanatory variables, *Y.na.mat* is the matrix of the extracted features, and $formula$ specifies the model we use, namely, model (2) in this example. For both the explanatory dataframe X and the response matrix Y, each row gives the values for different plants. The columns of X and *Y.na.mat* correspond to explanatory variables and observation dates, respectively. If a measurement of a plant is missing on a particular date, the value is filled by "NA" in the matrix.

4.2 Model Fitting

Based on the phenotypical parameters we obtained by processing images, we can study the relationship between the plant sizes and treatments, functional data models. Let $y_i(t)$ be the size of the i^{th} plant measured at time t, where $i = 1,...,9$. We treat genotype and block as fixed effects. There are 3 different genotypes and 3 blocks in this study. We can use $\mathbf{G}_i = (G_{ik})_{k=2}^3$ as the categorical indicator of the i^{th} plant genotype. Specifically, G_{ik} is set to one if the plant has the k^{th} genotype; otherwise, $G_{ik} = 0$. For example, if the i^{th} plant has the 2^{nd} genotype, then $\mathbf{G}_i = (0,1,0,...,0)$. And G_is being zeros represents the 1^{st} genotype, which is treated as the baseline. Similarly, $\mathbf{P_i} = (P_{ik})_{k=2}^3$ is defined to indicate the block that the i^{th} plant belongs to, and the first block is set as the baseline. Our functional ANOVA model for analyzing the relationship between plant size and genotype, block is:

$$y_i(t) = \mu(t) + \sum_{k=2}^{3} G_{ik} g_k(t) + \sum_{k=2}^{3} P_{ik} p_k(t) + \varepsilon_i(t), \tag{1}$$

where $\mu(t)$ is the growth function of the plant with the 1^{st} genotype (Genotype 78) from the 1^{st} block, $g_k(t)$s are the genotype effect functions, $p_k(t)$s are the fixed block effect functions, and the residuals $\varepsilon_i(t)$s are modeled by independent random processes with mean zeros. The observation time points in this study are irregular. Let m_i be the number of days that the i^{th} plant was imaged, which could vary from plant to plant. Let t_{ij} be the j^{th} observation time of the i^{th} plant, where $j = 1,...,m_i$. Then, our model can be described as:

$$y_i(t_{ij}) = \mu(t_{ij}) + \sum_{k=2}^{3} G_{ik} g_k(t_{ij}) + \sum_{k=2}^{3} P_{ik} p_k(t_{ij}) + \varepsilon_i(t_{ij}). \tag{2}$$

Since what we want is to study the relationship of the size and genotype effect, block effect, With all the input terms being prepared, we can fit the model using function *fanova*:

```
fit = fanova(Y.na.mat = Y.na.mat, X = X, tt = tt,
                  formula = "~ Genotype + Block",
                  K.int = 6, order = 4,
                  lower = -10, upper = 15)
```

More details can be found in the help documentation.

4.3 Statistical Inference

By fitting the model, we can draw statistical inference, such as parameter estimation, confidence bands, etc. For example, parameter estimation can be obtained by

```
para = fit$est_fun
```

where *para* is a t by q matrix. Here t is the number of observation time points, and q is the number of columns of the design matrix. In this case, $t = 23$, $q = 5$. The j^{th} column represents the estimated parameters of the j^{th} variable in the design matrix. For example, $\hat{\mu}(t)$ can be obtained by:

```
mu.t.hat = para[,1]
```

Afterwards, we can test the significance of the treatment effects of interest by constructing the corresponding confidence regions with the function "*CI_contrast*" in the package. For example, we want to test the difference between block 2 and block 1:

```
blk2_1 = CI_contrast(fit = fit, j1 = 4, j2 = 1, alpha = 0.05)
```

where fit is the output from "*fanova*", and $j1$ and $j2$ specify the columns of the design matrix corresponds to the treatment of interest. For example, the confidence region of the block effect function $p_2(\cdot)$ infers whether there is significant difference in plant size between block 1 and block 2, given the same plant genotype. Figure 2 (a) shows the 95% confidence regions of p_2 from day 1 to day 44. We find that, compared to block 1, block 2 does not have significant effect on the size of plants in the early stage of the growth until about the 15^{th} day.

In general, we can estimate a linear combination of treatment effects and its confidence regions by the following function:

```
CI(fit, L, alpha)
```

where L is a contrast vector under the model (2) specifying the linear combination of the parameters of interest. This includes estimating the average growth curve of a particular genotype over all the blocks. For example,

```
geno_2 = CI(fit = fit, L = c(1,0,0,1/3,1/3), alpha = 0.05)
```

Users can use $fit\$design\_mat$ from the output of the function $fanova$ to obtain the design matrix under this model, which will be helpful in defining the vector L. The design matrix of this problem is shown below:

```
(Intercept) Genotype2  Genotype3   Block2  Block3
    1          0           1         0       0
    1          0           0         0       0
    1          1           0         0       0
    1          1           0         1       0
    1          0           1         1       0
    1          0           0         1       0
    1          0           1         0       1
    1          1           0         0       1
    1          0           0         0       1
```

Users can identify L by reference to the design matrix.

5 FANOVA Example with an Interaction Term (Example 2)

Here is another example of fanova model with an interaction term. In this example, we use the data presented in (Xu et al., 2018). Data can be obtained by

```
data_Xu = read.csv(system.file("extdata", "data_Xu.txt",package = "implant",
                  mustWork = TRUE), sep = "")
```

Assuming that the image processing part has been done and we have obtained the extracted features. The object of this study is to model the growth curve of maize over time, using the plant area as the response variable, and using genotype and water treatment(well watered/water stressed) as explanatory variables. Moreover, we also consider the interaction between genotype and water treatment. Consider the genotype has two levels: FFMM-A maize and B73 maize, and the water treatment variable has two levels: well watered and water stressed. 60 plants were recorded daily for a total of 21 days, excluding day 16 as a result of a technical failure. Therefore, the model is written as:

$$y_i(t) = \mu(t) + G_i g(t) + W_i w(t) + G_i W_i \gamma(t) + \varepsilon_i(t) \tag{3}$$

where $y_i(t)$ is the area of the ith maize, and $\mu(t)$ is the growth function of water stressed FFMM-A maize. We denote $G_i = 1$ as the genotype of the ith maize is B73, 0 if the genotype of the ith maize is FFMM-A maize. Similarly, $W_i = 1$ means the ith maize is well watered, and 0 means the ith maize is water stressed. $g(t)$ and $w(t)$ are genotype effect and water effect, respectively. Moreover, $\gamma(t)$ is the interaction effect, and $\varepsilon_i(t)$ is the random process. We then add a penalty term of the second derivative of the basis expansion function to our model. We use function $fanova$ in our package to perform the model:

```
water = as.factor(data_Xu$water)
genotype = as.factor(data_Xu$genotype)
X = data.frame(water,genotype)
Y = data_Xu[,c(1:20)]
fit = fanova(Y.na.mat = Y, X = X,
                  tt = c(0:15,17:20),
                  formula = "~water + genotype + water:genotype",
                  K.int = 6, order = 4,
                  lower = -10, upper = 15)
```

We can obtain the estimated functional parameters using the output:

```
fit$est_fun
```

where the first, second, third and fourth columns of the *est_fun* matrix represent $\hat{\mu}(t)$, $\hat{g}(t)$, $\hat{w}(t)$, and $\hat{\gamma}(t)$, respectively. We can use other functions in this package, such as *CI_contrast*, to test the significance of each variable. For example, we can test whether the interaction term is significant by constructing its 95% confidence interval:

```
diff = CI_contrast(fit1, j1 = 4, j2 = 1, alpha = 0.05)
trt = diff$trt
upper_bound = diff$ub
lower_bound = diff$lb
```

Again, we can use the design matrix output by "fanova", that is,

```
fit$design_mat
```

to help us identify the values of j1 and j2. Te first six rows of the design matrix under the above model is listed below:

```
(Intercept)  water1      genotype1    water1:genotype1
    1           1           0               0
    1           0           0               0
    1           1           0               0
    1           0           0               0
    1           1           0               0
    1           0           0               0
```

It is easy to know that we are interested in the 4^{th} column and the 1^{st} column from the above matrix, and that's the reason we set $j1 = 4, j2 = 1$. Then we can draw a plot with respect to the confidence bands. See Figure 2 (c) for the confidence bands. The confidence bands indicates that we are 95% confident that the coefficient $\gamma(t)$ is within the confidence region. We also find that the confidence region of the coefficient contains 0, which means the interaction term is not significant. This implies that the treatment(watered/stressed) has the same effect on two genotypes. Therefore, we can drop this term. We fit the model without the interaction term and then we run the following code to see the growth curve over time t:

```
plot(tt,fit$est_fun[,1], type = "l", ylim = c(0,100000), lty = 2,col = "blue",
 xlab="Day",ylab="Plant Area (pixel)",cex.axis=1,cex.lab=1)
lines(tt,fit$est_fun[,1]+fit$est_fun[,2], lty = 2, col = "red")
lines(tt,fit$est_fun[,1]+fit$est_fun[,3], lty = 1,col = "blue")
lines(tt,fit$est_fun[,1]+fit$est_fun[,2]+fit$est_fun[,3], col = "red")
legend("topleft",col=c("red","red","blue","blue"),
lty=c(1,2,1,2),
legend =c("Watered,B73","Watered,FFMMA","Stressed,B73","Stressed,FFMMA"),
lwd=0.5,cex=0.8)
```

Comparing the result obtained by our package to the result obtained by (Xu et al., 2018), we found that they are exactly the same. See Figure 3 in the Appendix.

R Code for both examples can be found in the file *"demo.R"* from the *"data-raw"* folder of the package.

6 Appendix

References

Xu, Yuhang, Yumou Qiu, and James C Schnable (2018), "Functional modeling of plant growth dynamics." *The Plant Phenome Journal*, 1.

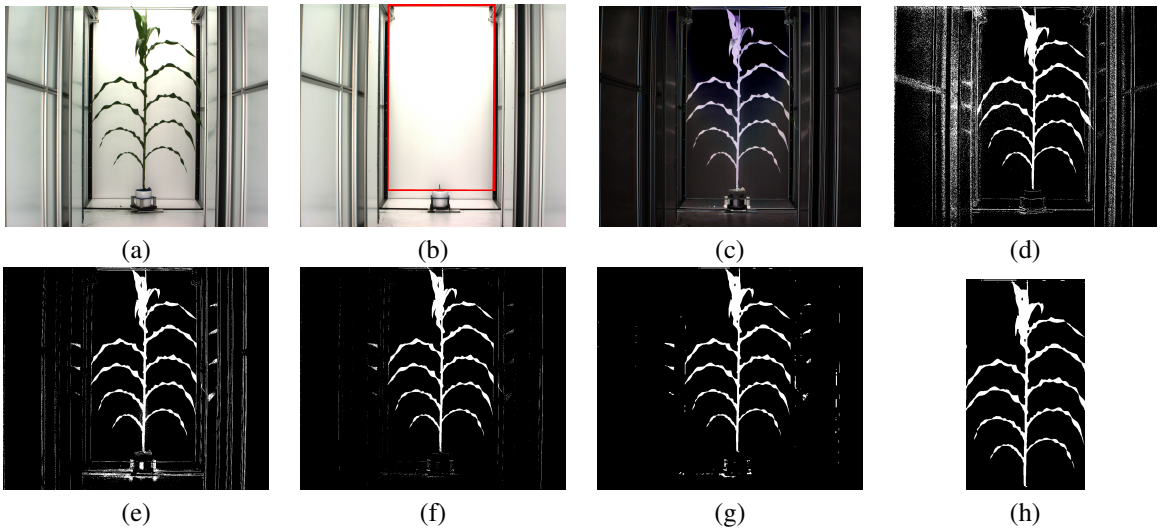
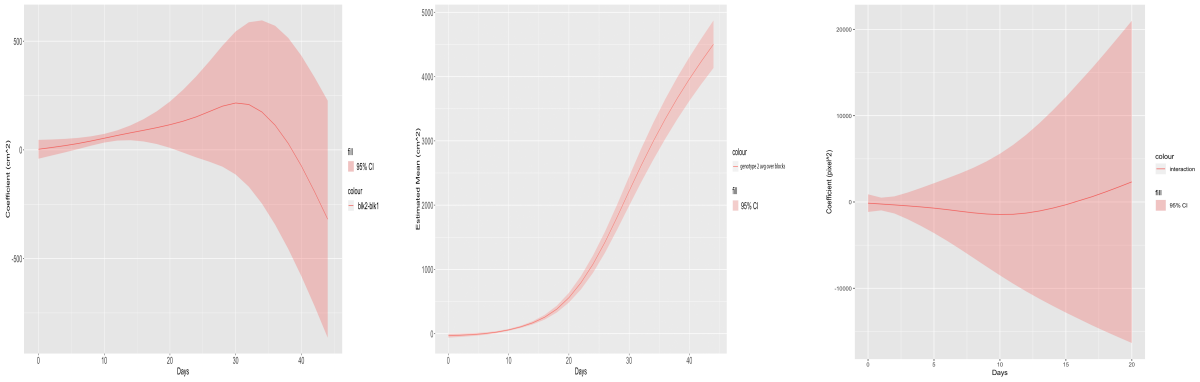
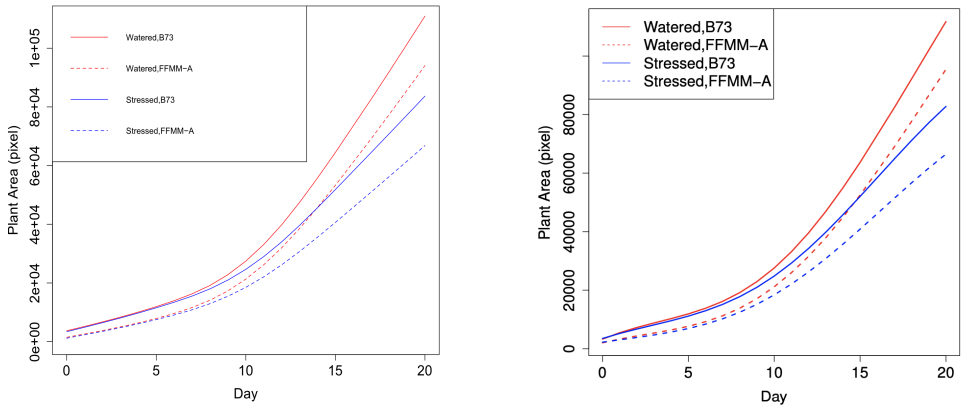


Figure 1. (a) Original plant image. (b) Original empty pot image; the red square is the identified region of interest by the functions *"ColorB"* and *"ColorG"*. (c) Contrast of (a) and (b). (d) Segmented image of (a) using DCT (e) Segmented image of (c) using DCT. (f) Intersection of (d) and (e). (g) Dilated-Eroded-Eroded-Dilated image of (f). (h) Final segmented image by identifying the region of interest.



(a) (b) (c)

Figure 2. (a) 95% confidence bands of the coefficient of the difference between block 2 and block 1. (b) 95% confidence bands of the estimated mean of genotype 2 averaging over three blocks. (c) 95% confidence bands of the interaction term of the Example 2



(a) (b)

Figure 3. (a) Growth curves estimated for each genotype x treatment combination obtained by the package ”implant”. (b) Growth curves estimated for each genotype x treatment combination in (Xu et al., 2018).