implant: Functional Data Analysis Pipeline for High Throughput Plant Phenotyping
User's Guide

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

High-throughput phenotyping system has become more and more popular in plant science research. The data analysis for such a system typically involves two steps: plant feature extraction through image processing and statistical analysis for the extracted features. The current approach is to perform those two steps on different platforms. We develop the package *"implant"* in R for both robust feature extraction and functional data analysis, which is able to provide statistical inference for the plant traits directly from the input images. For image processing, the *"implant"* package provides methods including thresholding, hidden Markov random field model, morphology operations and etc. For the growth curve analysis, this package can produce nonparametric curve fitting with its confidence region for plant growth. A functional ANOVA model to test for the treatment and genotype effects of the growth curve dynamics is also provided.

Contents

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 analysis part, we use the data extracted from the processed images in the first part, and use the functional anova approach to build models and have the corresponding analysis.

2 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.

2.1 Defining the Region of Interest

Using the package "*png*", we can transform these two images into matrices of pixel intensities. Let's name Figure 1(a) as *imageOriginal* and Figure 1(b) as *image*2. 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 4. Then we can check the dimensions of these two matrices. For example, we'd like to find the dimensions for *imageOriginal* for each channel:

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

And we can see that in each channel, the matrix has 2056 rows and 2454 columns. Using these information, we can define our region of interest.

Using the function *ColorG*, we can find the green strip from *image*2 and change its color from green to the background color, white. Also we can find the vertical location of the top of the pot (the location is the same as the location is the bottom part of the plant body in Figure 1(a).)

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

We may also reduce the sizes of these two images using function *sample*, and name the reduced original image and the reduced empty pot image as iamge1 and imageX1, respectively.

```
image1 = sample(imageOriginal, RowSample = 2, ColSample = 2)
imageX1 = sample(image2, RowSample = 2, ColSample = 2)
```

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

In order to obtain a well segmented result using thresholding method, we nned to identify the region of interest and change the color of the region that we are not interested in into black (Since we will segment the image in the next step and make the segmented image binary, which means coloring the background as black and the object of interest as white.). Using "*ColorB*" and "*ColorG*, we can identify the inner black bars and the border-top of the pot to obtain the region of interest for the plants. See the red rectangle in Figure 1 panel (b).

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

```
RI[c(lowerRowBound : sizeOrignal[1]), ] = 0
#Cut the image using the identified reigon of interest
imageOriginalR = imageOriginal[, , 1]
imageOriginalG = imageOriginal[, , 2]
imageOriginalB = imageOriginal[, , 3]
imageCutR = matrix(imageOriginalR[RI == 1],lowerRowBound - 1,
                   ColBound[2] - ColBound[1] - 1)
imageCutG = matrix(imageOriginalG[RI == 1],lowerRowBound - 1,
                   ColBound[2] - ColBound[1] - 1)
imageCutB = matrix(imageOriginalB[RI == 1],lowerRowBound - 1,
                   ColBound[2] - ColBound[1] - 1)
imageCut = array(c(imageCutR, imageCutG, imageCutB), dim = c(dim(imageCutR), 3))
```

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.

2.2 Image Segmentation using Thresholding Method

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 image1 and imageX1G (see 1 (c) in the Appendix).

```
tempG1 = ColorG(imageX1, rowThreshold = 0.002)
imageX1G = tempG1$c
imageContrast = abs(image1 - 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 image1 and imageContrast, respectively. Then we can have a well segmented result by finding the intersection between the segmented image1 and the segmented imageContrast. See Figure 1 (d) - (f) in the Appendix.

```
imageB1 = imageBinary(image1,  weight = c(-1, 2, -1), threshold1 = 30 / 255,
threshold2 = 0.02)
imageB2 = imageBinary(imageContrast1,  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_EM"*.

2.3 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) )
```

2.4 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 our 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 = sampleMatrix(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.

2.5 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 $1.549mm$ in horizontal and $1.553mm$ 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(processed_image = imageBF,
        Xsize = 1.54902601242065, Ysize = 1.55327594280243, a = 2, b = 2)$height
extract(processed_image = imageBF,
        Xsize = 1.54902601242065, Ysize = 1.55327594280243, a = 2, b = 2)$width
extract(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.

3 Functional ANOVA

3.1 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 define the input terms as follows:

```
data_new = read.csv("~/Desktop/sideview/new_data.csv",header = TRUE)
Y.na.mat = data_new[,-c(1:3)]
time_interval = seq(from = 0, to = 44,by = 2)
X = data_new[,c(2:3)]
#This step is to factorize each factor
X1 = as.factor(unlist(X[,1]))
X2 = as.factor(unlist(X[,2]))
X = data.frame(X1,X2)
formula = toString("~ X[,1]+X[,2]")
```

where X is a matrix 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 covariate matrix X and the response matrix $Y.na.mat$, 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. More details can be found in the help documentation.

3.2 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_mean*:

```
fit = fanova_mean(Y.na.mat, X = X, tt = time_interval, K.int = 6, order = 4,
                  interact = 0, p = 2, d0 = 0, d1 = 2, d2 = 2,
                  lower = -10, upper = 15, formula = formula)
```

3.3 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.trt.diff*" in the package. For example, we want to test the difference between block 2 and block 1:

```
blk2_1 = CI.trt.diff(fit = fit, j1 = 2, j2 = 1, alpha = 0.05)
```

where *fit* is the output from "*fanova_mean*", 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 $p_3(\cdot)$ infers whether there is significant difference in plant size between block 1 and block 3, given the same plant genotype. Figure 2 (a) shows the 95% confidence regions of p_3 from day 1 to day 44. We find that, compared to block 1, block 3 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.trt(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_78 = CI.trt(fit = fit, L = c(1,0,0,1/3,1/3), alpha = 0.05)
```

4 Appendix

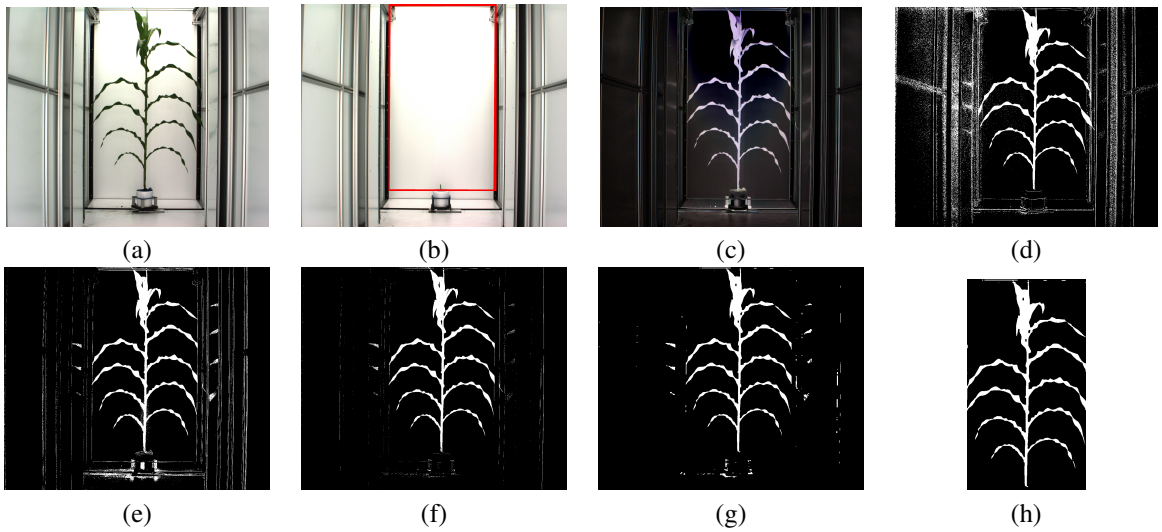
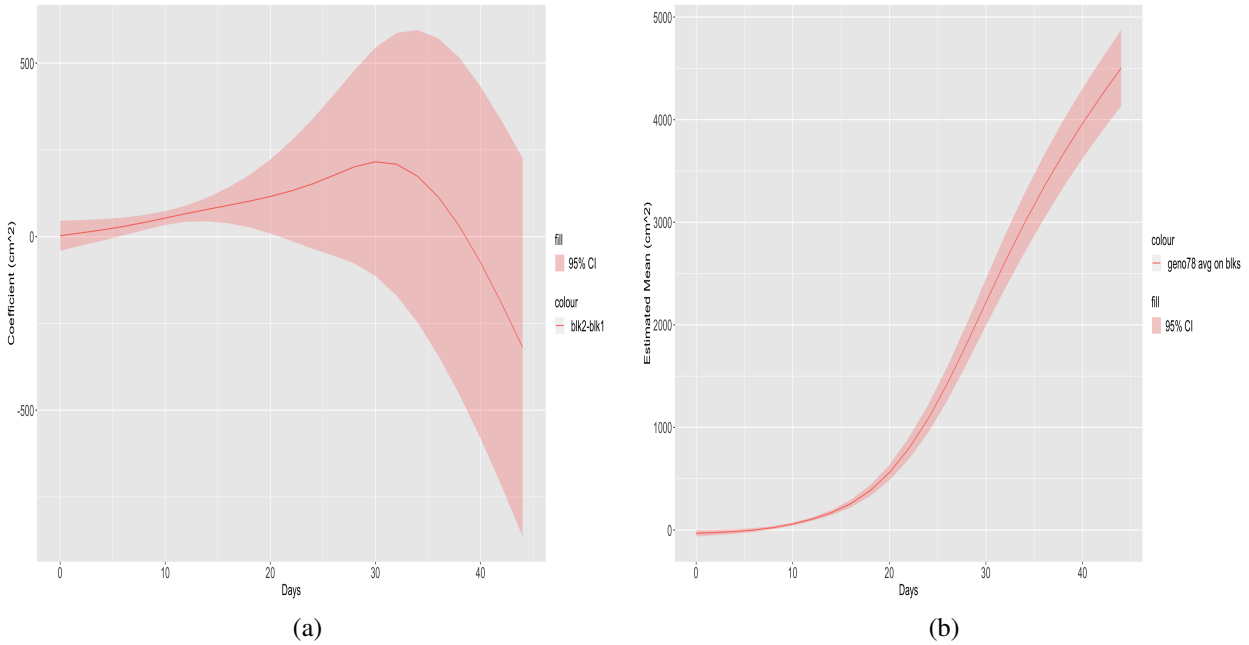


Figure 1. (a) Original plant image. (b) Original empty pot image; the red square is the identified region of interest by the functions *"ColorB"* and *"ColorG"*. (c) Contrast of (a) and (b). (d) Segmented image of (a) using DCT (e) Segmented image of (c) using DCT. (f) Intersection of (d) and (e). (g)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



(a) (b)

Figure 2. (a) 95% confidence band of the coefficient of the difference between block 2 and block 1. (b) 95% confidence band of the estimated mean of genotype 78 averaging over three blocks.