agriculture

MDPI

Supplementary Information

Probing Differential Metabolome Responses among Wheat Genotypes to Heat Stress using Fourier Transform Infrared-based Chemical Fingerprinting

Salma O. M. Osman [1,2], Abu Sefyan I. Saad [2], Shota Tadano [1], Yoshiki Takeda [3], Yuji Yamasaki [4], Izzat S. A. Tahir [2], Hisashi Tsujimoto [4], and Kinya Akashi [1,3,4,*]

[1] United Graduate School of Agricultural Sciences, Tottori University, 4-101 Koyama-Cho-Minami, Tottori 680-0945, Japan; salmamohamedkhair@gmail.com (S.O.M.O.); shota.tdn@gmail.com (S.T.)
[2] Agricultural Research Corporation, Wad Medani P.O. Box 126, Sudan; sefian_ib@yahoo.com (A.S.I.S.); izzatahir@yahoo.com (I.S.A.T.)
[3] Faculty of Agriculture, Tottori University, 4-101 Koyama-Chou-Minami, Tottori 680-0945, Japan; b18a5097c@edu.tottori-u.ac.jp
[4] Arid Land Research Center, Tottori University, 1390 Hamasaka, Tottori 680-0001, Japan; yujyamas@tottori-u.ac.jp (Y.Y.); tsujim@tottori-u.ac.jp (H.T.)
* Correspondence: akashi.kinya@tottori-u.ac.jp; Tel./Fax: +81-857-31-5352

Supplemental Table S1. One-way ANOVA with post-hoc Tukey HSD test on canopy temperature.

Category	Pair	Tukey HSD Q statistic	Tukey HSD *p* -value *
CS	C0 vs C3	0.5469	0.7194
	C0 vs H3	11.0873	0.001
	C3 vs H3	14.0014	0.001
Imam	C0 vs C3	1.1828	0.4225
	C0 vs H3	26.2122	0.001
	C3 vs H3	11.2751	0.001
N61	C0 vs C3	0.113	0.85
	C0 vs H3	17.9206	0.001
	C3 vs H3	24.0925	0.001
C0	CS vs Imam	1.662	0.2671
	CS vs N61	1.066	0.4683
	Imam vs N61	0.6218	0.6823
C3	CS vs Imam	1.722	0.2513
	CS vs N61	0.6856	0.6507
	Imam vs N61	1.427	0.3368
H3	CS vs Imam	2.052	0.1775
	CS vs N61	0.5831	0.7014
	Imam vs N61	1.701	0.2567

* The *p*-values <0.5 and >0.5 are indicated by light green and pink colors, respectively.

Supplemental Table S2. One-way ANOVA with post-hoc Tukey HSD test on total leaf length.

Category	Pair	Tukey HSD Q statistic	Tukey HSD *p* -value *
CS	C0 vs C3	8.557	0.001
	C0 vs H3	2.236	0.145
	C3 vs H3	5.801	0.0021
Imam	C0 vs C3	8.495	0.001
	C0 vs H3	8.495	0.001
	C3 vs H3	3.845	0.0216
N61	C0 vs C3	19.3	0.001
	C0 vs H3	6.28	0.0013
	C3 vs H3	4.856	0.0064
C0	CS vs Imam	2.743	0.0811
	CS vs N61	3.002	0.06
	Imam vs N61	0.0964	0.85
C3	CS vs Imam	3.465	0.0342
	CS vs N61	5.815	0.0021
	Imam vs N61	1.778	0.2372
H3	CS vs Imam	0.0239	0.85
	CS vs N61	0.7883	0.6
	Imam vs N61	1.404	0.3443

* The *p*-values <0.5 and >0.5 are indicated by light green and pink colors, respectively.

Supplemental Table S3. One-way ANOVA with post-hoc Tukey HSD test on shoot biomass.

Category	Pair	Tukey HSD Q statistic	Tukey HSD *p* -value *
CS	C0 vs C3	0.001	0.001
	C0 vs H3	3.567	0.0302
	C3 vs H3	4.158	0.0148
Imam	C0 vs C3	7.614	0.001
	C0 vs H3	8.352	0.001
	C3 vs H3	3.384	0.0378
N61	C0 vs C3	10.51	0.001
	C0 vs H3	4.279	0.0143
	C3 vs H3	4.233	0.0135
C0	CS vs Imam	2.679	0.0874
	CS vs N61	0.1965	0.8929
	Imam vs N61	0.1965	0.8929
C3	CS vs Imam	1.107	0.4518
	CS vs N61	3.033	0.0576
	Imam vs N61	4.147	0.015
H3	CS vs Imam	3.2572	0.0861
	CS vs N61	0.5501	0.9
	Imam vs N61	3.8073	0.0417

* The *p*-values <0.5 and >0.5 are indicated by light green and pink colors, respectively.

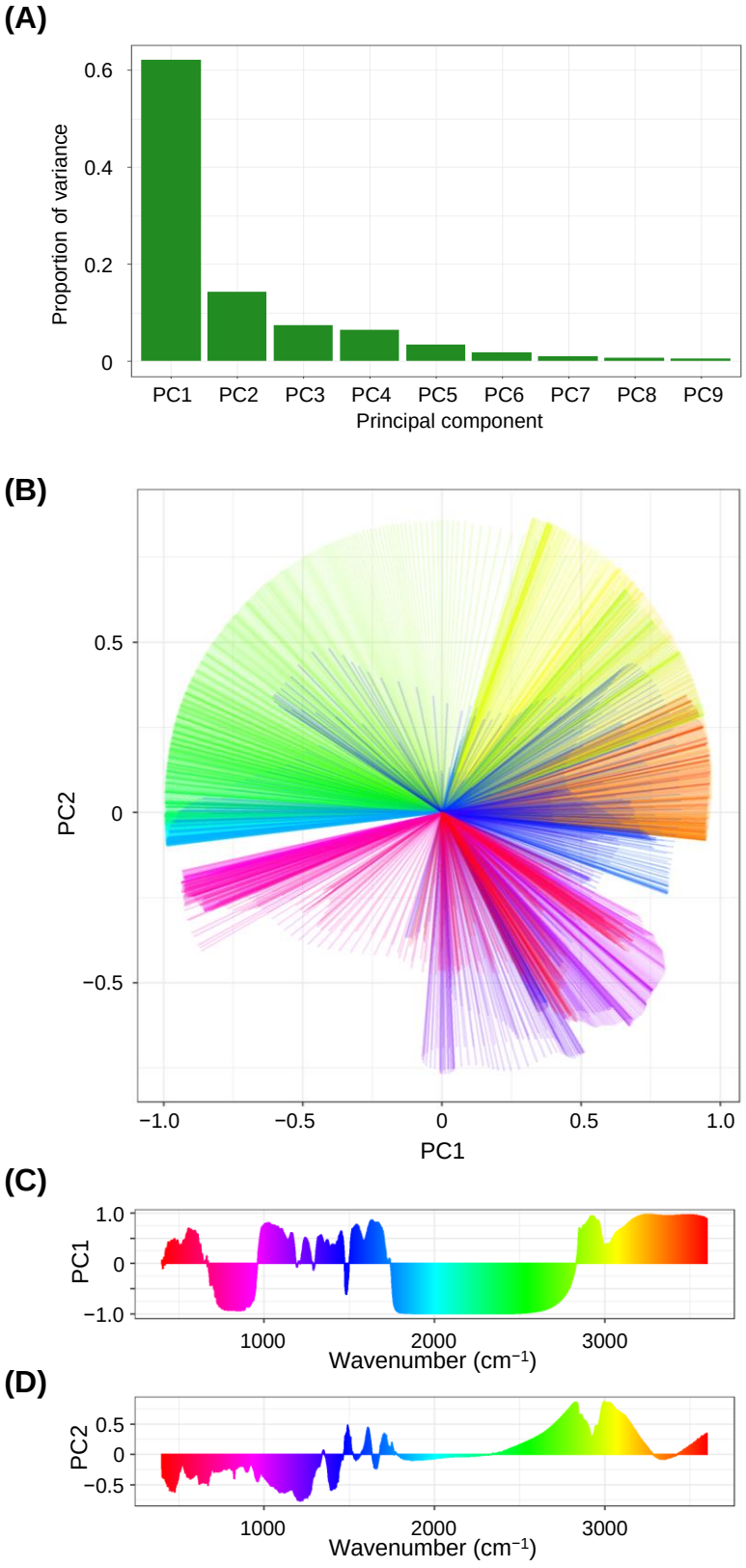
Supplementary Table S4. Characteristics of spectral markers.

Marker	Wavenumber (cm^{-1})[*1]			CS			Imam			N61		
	Target	Anchor 1	Anchor 2	Median Fm value		p [*3]	Median Fm value		p [*3]	Median Fm value		p [*3]
				C3 [*2]	H3 [*2]		C3 [*2]	H3 [*2]		C3 [*2]	H3 [*2]	
Fm482	482	401	501	0.8587	0.7629	2.0×10^{-9}	0.8186	0.7788	4.3×10^{-7}	0.7401	0.6601	1.9×10^{-9}
Fm576	576	648	542	1.0621	1.3538	7.1×10^{-5}	1.1481	1.1231	8.8×10^{-2}	1.3327	0.9074	9.0×10^{-8}
Fm1251	1251	1241	1358	-0.1770	-0.0309	1.6×10^{-7}	-0.0473	-0.0915	1.6×10^{-3}	-0.0416	-0.1318	1.0×10^{-7}
Fm1465	1465	1480	1399	0.2921	0.3351	3.8×10^{-3}	0.4268	0.3806	1.9×10^{-4}	0.3463	0.3817	2.5×10^{-13}
Fm1502	1502	1480	1615	0.3381	0.2770	1.2×10^{-18}	0.3151	0.3115	1.8×10^{-4}	0.3328	0.2919	4.4×10^{-15}
Fm1729	1729	1768	1703	0.6291	0.6217	1.8×10^{-1}	0.5861	0.6040	1.1×10^{-6}	0.5615	0.5895	2.0×10^{-16}

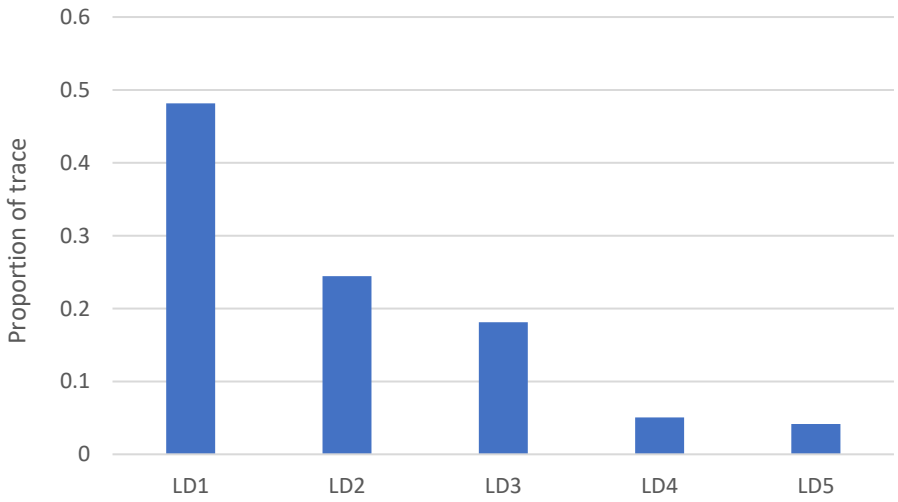
[1] Wavenumbers for the target and the flanking two anchors, that were used for the Fm value calculation as described previously [35].

[2] Fm value in either C3 (control day 3) or H3 (heat day 3) condition.

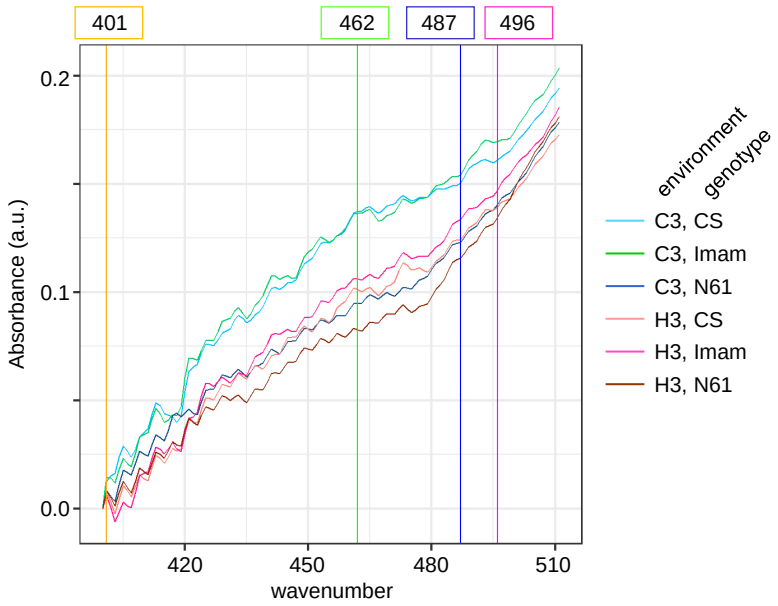
[3] Significance between C3 and H3 conditions by *t* -test.



Supplementary Figure S1. Supplementary data on the PCA of FTIR spectra. (A) Variance explained by the first nine components in PCA. (B) Two-dimensional PC1-PC2 loading plot. (C, D) One dimensional loading plot for (C) PC1 and (D) PC2. The colors of the vectors in panel (B) are the same as those in the panels (C) and (D).



Supplementary Figure S2. Values of proportion of trace in linear discriminant analysis.



Supplementary Figure S3. Magnified view of averaged FTIR spectra in the wavenumber ranges from 400 to 510 cm^{-1} that detected strong discriminatory variable wavenumbers in LDA. Averaged spectra for 6 genotypes × environment combinations were drawn as the color legend as depicted in the right of the panel. Colored vertical straight lines and their numbers on top of the panel denote the characteristic wavenumbers that were detected as strong discriminatory variable in LDA.

Supplementary Document S1.
Osman *et. al.*
R-scripts for the processing of FT-IR data

Script code 1: FTIR-spectra processing

```
#salma c2.1a spec processing with wn400-4000 offset baseline
#import necessary libraries
library(conflicted)
library(tidyverse)

#clean up the R's brain
rm(list=ls())

#obtain date information
today <- Sys.Date()
yr <- substr(today, 3,4)
mo <- substr(today, 6,7)
day <- substr(today, 9,10)
today2 <- paste(yr, mo, day, sep="")

# !!!    system check required, 1 out of 2
#obtain desktop folder information for a windows user
#you must change the string below within "xxx" according to your computer
username <- "akash"

#prepare output folder and its path
DesktopPath <- paste("C:/users/",username,"/desktop/", sep="")
setwd(DesktopPath)
if(!dir.exists(paste(today2, "_specpile/", sep=""))){
   dir.create(paste(today2, "_specpile/", sep=""), recursive=T)
}
OutputPath <- paste(DesktopPath, today2, "_specpile/", sep="")

# !!!    system check required, 2 out of 2
#prepare input data folder
#subfolder below the "rawdata" will be ignored
setwd("d:/1_DataFolder/Intel/i04_Informatics_Statistics/i04b_R/trainingdata/ftir_testdata/salma
_testdata/paper3/rawdata/220330_FTIR_rawdata_c3_h3/")
#setwd("d:/1_DataFolder/Intel/i04_Informatics_Statistics/i04b_R/trainingdata/ftir_testdata/salm
a_testdata/paper3/rawdata/220327_ftir_rawdata_5dates/")
#setwd("d:/1_DataFolder/Intel/i04_Informatics_Statistics/i04b_R/trainingdata/ftir_testdata/salm
a_testdata/paper3/rawdata/01_selected")
#setwd("d:/1_DataFolder/Intel/i04_Informatics_Statistics/i04b_R/trainingdata/ftir_testdata/salm
a_testdata/paper3/rawdata/02_newdata")
pathname_inputfolder <- getwd()
pathname_inputfolder

#create column names for output dataframe
wnlist <- seq(4000, 400, length=3601)
columnname <- c("filename", "condition", "genotype", "identifier", wnlist)

#name the column labels for spec data
specpile <- as.data.frame(t(columnname))
```

```
names(specpile) <- columnname
specpile <- slice(specpile, -1)

#prepare dataframe for NG spec
NGspec <- specpile

#input data from .asc file that are generated by Perkin-Elmer
#obtain the filename
#obtain the list of filenames for all csv files,
   #which are stored in the aforementioned "rawdata" folder
filelist <- list.files(path = pathname_inputfolder,
                                    pattern = "*.asc",
                                    full.names = T)

#count the number of files
fileno <- length(filelist)

#starting a loop for processing data
for (i in 1:fileno){
#for (i in 1:1){

  #obtain the new filename
       filename <- basename(filelist[i])

  #obtain dataframe
  #skip first 25 lines
  #the 26th line does not have variable names
    rawspec <- read.table(filelist[i], skip = 25)

  #quick summary
#    summary(rawspec)

  #plot the spectrum
#    ggplot(rawspec, aes(x = V1,y = V2)) +
#      geom_point()

  #save the wn column for later plotting
    wn_column <- dplyr::select(rawspec, V1)

  #exchange rows and columns
  #(optional)keep the type as data.frame
    rawspec2 <- as.data.frame(t(rawspec))

  #split the rows into wn and spec
    wn_axis <- as.data.frame(rawspec2[1,])
    rawspec3 <- as.data.frame(rawspec2[2,])

  #name the column labels for spec data
    names(rawspec3) <- wn_axis

    #smoothing of the spectrum trace
    #below to fill in, but currently skip it

    #obtain the baseline anchors
```

```
#this is a version to take only 4000 and 400
#the relationships between wn and column_no. is
#column_no = -wn +4001
raw400 <- rawspec3[1,3601]
raw4000 <- rawspec3[1,1]

#create a baseline data
#following is the 1st version
#line is drawn between 4000 to 400
baseline <- seq(raw4000, raw400, length=3601)

#subtract the baseline
spec4_baselined <- (rawspec3 - baseline)

#draw the baseline-corrected spectrum
#1st, exchange the rows and columns
spec4_tall <- t(spec4_baselined)

#combine the wn and spec columns
spec4_tall <- cbind(wn_column, spec4_tall)

#plot the baseline-corrected spectrum
#ggplot(spec4_tall, aes(x = V1,y = V2)) +
#    geom_point(size=0.3)

#normalization of spec
#1st, sum of current spec is calculated
sum_signal_original <- sum(dplyr::select(spec4_tall, V2))

#2nd, new column is generated in the spec
#spec values in ppm is calculated
spec5_tall <- dplyr::mutate(spec4_tall, ABS = V2*1000000/sum_signal_original)

#draw the normalized spectrum
#ggplot(spec5_tall, aes(x = V1,y = ABS)) +
#    geom_point(size=0.3)

#row-column conversion
spec5 <- as.data.frame(t(spec5_tall))

#remove original data from spec5
spec6 <- dplyr::slice(spec5, 3)

#rownames(spec6) <- filename

#create one column at the top
#add dataname to the 1st column
spec7 <-    mutate(spec6, dataname=filename, .before="4000")

#extract treatment condition info, and add to the 2nd column
condition_info <- substring(filename, 1, 2)
spec8 <- mutate(spec7, condition=condition_info, .after="dataname")

#extract genotype info, and add to the 3rd column
```

```
  genotype_info <- substring(filename, 3, 5)
  spec9 <- mutate(spec8, genotype=genotype_info, .after="condition")

  #setup the identifier for later analyses
  #and add to the 4th column
  identifier_info <- paste(condition_info, genotype_info, sep="")
  spec10 <- mutate(spec9, identifier=identifier_info, .after="genotype")

  #eliminate, if any, NG data taken by transmission-mode,
    #and maintain only OK data with absorption-mode
    #the OK data has higher absorbanse at wn3400 than at wn1800
    #these wn correspond to col607 and col2207, respectively, in spec10
  #if(spec10[1,607]>spec10[1,2207]){
    #compiling the data
    specpile <- rbind(specpile, spec10)
  #} else {
  #   NGspec <- rbind(NGspec, spec10)
  #}

  }

#export the data as csv
#data is baseline-corrected, normalized spec
setwd(OutputPath)
a <- getwd()
a
filename_specpile_processed <- paste(today2, "_c2.1a_specpile_offsetbaselined.csv", sep="")
write.csv(specpile,
              filename_specpile_processed, row.names=FALSE)

#export NG data as well
filename_NGspec_processed <- paste(today2, "_c2.1a_NGspecpile_offsetbaselined.csv", sep="")
write.csv(NGspec,
              filename_NGspec_processed, row.names=FALSE)

#prepare long-format as well, and export
#row-column conversion
long_specpile <- as.data.frame(t(specpile))
long_NGspec <- as.data.frame(t(NGspec))

#create new column at the top
long_specpile <- mutate(long_specpile,
        variable=c("dataname","condition", "genotype", "identifier",
                        seq(from=4000, to=400, by=-1)),
                              .before=ABS)
long_NGspec <- mutate(long_NGspec,
        variable=c("dataname","condition", "genotype", "identifier",
                        seq(from=4000, to=400, by=-1)),
                                    .before=ABS)

#export the data as csv
#data is baseline-corrected, normalized spec
filename_specpile_processed_longformat <- paste(today2, "_c2.1a_speclong_offsetbaselined.csv",
sep="")
```

```
write.csv(long_specpile,
          filename_specpile_processed_longformat, row.names=FALSE)
filename_NGspec_processed_longformat <- paste(today2,
"_c2.1a_NGspeclong_offsetbaselined.csv", sep="")
write.csv(long_NGspec,
          filename_NGspec_processed_longformat, row.names=FALSE)
```

Script code 2: Averaging spectra

```
#c3.1a spec averaging ftir
   #for averaging the spectra
   #This version of script is specific to salma-3 paper
      #for genotype comparison in c3 and h3 condition.

#input file is offset-baselined "specpile_processed.csv"

#import necessary libraries
library(conflicted)
library(tidyverse)
library(psych)

#clean up the R's brain
rm(list=ls())

#obtain date information
today <- Sys.Date()
yr <- substr(today, 3,4)
mo <- substr(today, 6,7)
day <- substr(today, 9,10)
today2 <- paste(yr, mo, day, sep="")

# !!!   system check required, 1 out of 2
#obtain desktop folder information for a windows user
#you must change the string below within "xxx" according to your computer
username <- "akash"

#prepare output folder and its path
DesktopPath <- paste("C:/users/",username,"/desktop/", sep="")
setwd(DesktopPath)
if(!dir.exists(paste(today2, "_drawspec/", sep=""))){
   dir.create(paste(today2, "_drawspec/", sep=""), recursive=T)
}
OutputPath <- paste(DesktopPath, today2, "_drawspec/", sep="")

# !!!   system check required, 2 out of 2
#prepare input data folder
setwd("d:/1_DataFolder/Intel/i04_Informatics_Statistics/i04b_R/trainingdata/ftir_testdata/salma
_testdata/paper3/")
a <- getwd()
a

#import the 1st, compiled ftir csv data
print("Please specify c2.1a_specpile_offsetbaselined.csv")
specpile1 <- file.choose()
```

```
specpile2 <- read.csv(specpile1, header = T)

#split the data according to their identifiers
c3chs_specpile2 <- dplyr::filter(specpile2, identifier == 'c3chs')
c3ima_specpile2 <- dplyr::filter(specpile2, identifier == 'c3ima')
c3n61_specpile2 <- dplyr::filter(specpile2, identifier == 'c3n61')
h3chs_specpile2 <- dplyr::filter(specpile2, identifier == 'h3chs')
h3ima_specpile2 <- dplyr::filter(specpile2, identifier == 'h3ima')
h3n61_specpile2 <- dplyr::filter(specpile2, identifier == 'h3n61')

#separate the identifier column (category info)
c3chs_id_1 <- dplyr::select(c3chs_specpile2, (1:4))
c3ima_id_1 <- dplyr::select(c3ima_specpile2, (1:4))
c3n61_id_1 <- dplyr::select(c3n61_specpile2, (1:4))
h3chs_id_1 <- dplyr::select(h3chs_specpile2, (1:4))
h3ima_id_1 <- dplyr::select(h3ima_specpile2, (1:4))
h3n61_id_1 <- dplyr::select(h3n61_specpile2, (1:4))

#extract values used for averaging
c3chs_specmatrix <- dplyr::select(c3chs_specpile2, -(1:4))
c3ima_specmatrix <- dplyr::select(c3ima_specpile2, -(1:4))
c3n61_specmatrix <- dplyr::select(c3n61_specpile2, -(1:4))
h3chs_specmatrix <- dplyr::select(h3chs_specpile2, -(1:4))
h3ima_specmatrix <- dplyr::select(h3ima_specpile2, -(1:4))
h3n61_specmatrix <- dplyr::select(h3n61_specpile2, -(1:4))

#averaging
c3chsmean <- as.data.frame(t(apply(c3chs_specmatrix, 2, mean)))
c3imamean <- as.data.frame(t(apply(c3ima_specmatrix, 2, mean)))
c3n61mean <- as.data.frame(t(apply(c3n61_specmatrix, 2, mean)))
h3chsmean <- as.data.frame(t(apply(h3chs_specmatrix, 2, mean)))
h3imamean <- as.data.frame(t(apply(h3ima_specmatrix, 2, mean)))
h3n61mean <- as.data.frame(t(apply(h3n61_specmatrix, 2, mean)))

specmean <- rbind(c3chsmean, c3imamean, c3n61mean,
                  h3chsmean, h3imamean, h3n61mean)
rownames(specmean) <- c("c3chs", "c3ima", "c3n61",
                        "h3chs", "h3ima", "h3n61")

#save the specmean as csv file
setwd(OutputPath)
tempa <- getwd()
tempa
filename_specmean <- paste(today2, "_c3.1a_specmean_offset.csv", sep="")
write.csv(specmean,
          filename_specmean, row.names=TRUE)

#save a long-format of specmean as csv file
specmeanlong <- as.data.frame(t(specmean))
wn <- seq(from=4000, to=400, by=-1)
wn_col <- as.data.frame(wn)
specmeanlong2 <- cbind(wn_col, specmeanlong)
filename_specmeanlong <- paste(today2, "_c3.1a_specmeanlong_offset.csv", sep="")
write.csv(specmeanlong2,
```

```
filename_specmeanlong, row.names=TRUE)
```

Script code 3: Drawing entire spectra

```
#salma_c3.2_draw_spec
#this is to draw averaged or representative spectra for genotype paper

#import necessary libraries
library(conflicted)
library(tidyverse)
library(caret)
library(ggpubr)

#clean up the R's brain
rm(list=ls())

#obtain date information
today <- Sys.Date()
yr <- substr(today, 3,4)
mo <- substr(today, 6,7)
day <- substr(today, 9,10)
today2 <- paste(yr, mo, day, sep="")

# !!!    system check required, 1 out of 2
#obtain desktop folder information for a windows user
#you must change the string below within "xxx" according to your computer
username <- "akash"

#prepare output folder and its path
DesktopPath <- paste("C:/users/",username,"/desktop/", sep="")
setwd(DesktopPath)
if(!dir.exists(paste(today2, "_drawspec/", sep=""))){
   dir.create(paste(today2, "_drawspec/", sep=""), recursive=T)
}
OutputPath <- paste(DesktopPath, today2, "_drawspec/", sep="")

# !!!    system check required, 2 out of 2
#prepare input data folder
setwd("d:/1_DataFolder/Intel/i04_Informatics_Statistics/i04b_R/trainingdata/ftir_testdata/salma
_testdata/paper3/")
pathname_inputfolder <- getwd()
pathname_inputfolder

#invoke a file-opening window, specify the input file,
#the input data should be in .csv format,
   #and has to be long-format, with baseline-corrected and normalized
#obtain the filename
print("Please specify xxxxxx_c3.1a_specmeanlong_offset.csv")
print("or, xxxxxx_c3.1b_specmeanlong_piecewise.csv")
print("or, xxxxxx_c3.1a_specrepresentativelong_offsetbaselined.csv")
longinputfile <- file.choose()
filename <- basename(longinputfile)
speclong6 <- read.csv(longinputfile,
                      header = T, row.names="X")
```

```
filename

#Session 1
#drawing 18 spectra in a single panel

#arrange the spectra data
#twospec2 <- dplyr::select(twospec, -(c(1:1)))
#colnames(twospec2) <- seq(from=4000, to=400, by=-1)

#wnlist1 <- as.data.frame(t(seq(from=4000, to=400, by=-1)))
#colnames(wnlist1) <- seq(from=4000, to=400, by=-1)
#twospec3 <- rbind(wnlist1, twospec2)

#c3id <- twospec[1,6]
#h3id <- twospec[2,6]
#longspec3 <- as.data.frame(t(twospec3))
#colnames(longspec3) <- c("wn", "c3", "h3")

#draw the overlapped spec
dev.new()
plotspec6 <- ggplot(speclong6) +
   theme_bw()+
   geom_line(aes(x=speclong6[,1], y=speclong6[,2]),
                     colour="deepskyblue", size=0.3)+
   geom_line(aes(x=speclong6[,1], y=speclong6[,3]),
                  colour="dodgerblue", size=0.3)+
   geom_line(aes(x=speclong6[,1], y=speclong6[,4]),
                  colour="dodgerblue4", size=0.3)+
   geom_line(aes(x=speclong6[,1], y=speclong6[,5]),
                  colour="salmon", size=0.3)+
   geom_line(aes(x=speclong6[,1], y=speclong6[,6]),
                  colour="salmon3", size=0.3)+
   geom_line(aes(x=speclong6[,1], y=speclong6[,7]),
                  colour="orangered4", size=0.3)+
   xlab("wavenumber")+
   ylab("ABS")+
   ggtitle(paste(today2,"_6spec",sep=""))
print(plotspec6)

#save the plot as png format(you can change to .jpeg, .tiff, etc)
setwd(OutputPath)
b <- getwd()
b
filename_plotspec6 <- paste(today2, "_c3.2_6spec.png", sep="")
ggsave(file = filename_plotspec6,
          plot = plotspec6, dpi=100,
          width=7.2, height=2.4)

#Session 2
#draw the upward-stacked spec
speclong6a <- speclong6
speclong6a <- dplyr::mutate(speclong6a, c3chs=c3chs+400)
speclong6a <- dplyr::mutate(speclong6a, c3ima=c3ima+800)
speclong6a <- dplyr::mutate(speclong6a, c3n61=c3n61+1200)
```

```
speclong6a <- dplyr::mutate(speclong6a, h3chs=h3chs+1600)
speclong6a <- dplyr::mutate(speclong6a, h3ima=h3ima+2000)
speclong6a <- dplyr::mutate(speclong6a, h3n61=h3n61+2400)

#draw the stacked spec
dev.new()
plotspec6a <- ggplot(speclong6a) +
   theme_bw()+
   geom_line(aes(x=speclong6a[,1], y=speclong6a[,2]),
                colour="deepskyblue", size=0.3)+
   geom_line(aes(x=speclong6a[,1], y=speclong6a[,3]),
                colour="dodgerblue", size=0.3)+
   geom_line(aes(x=speclong6a[,1], y=speclong6a[,4]),
                colour="dodgerblue4", size=0.3)+
   geom_line(aes(x=speclong6a[,1], y=speclong6a[,5]),
                colour="salmon", size=0.3)+
   geom_line(aes(x=speclong6a[,1], y=speclong6a[,6]),
                colour="salmon3", size=0.3)+
   geom_line(aes(x=speclong6a[,1], y=speclong6a[,7]),
                colour="orangered4", size=0.3)+
   xlab("wavenumber")+
   ylab("ABS")+
   ggtitle(paste(today2,"_6stacked_spec",sep=""))
print(plotspec6a)

#save the plot as png format(you can change to .jpeg, .tiff, etc)
setwd(OutputPath)
b <- getwd()
b
filename_plotspec6a <- paste(today2, "_c3.2_6spec_stacked.png", sep="")
ggsave(file = filename_plotspec6a,
          plot = plotspec6a, dpi=100,
          width=7.2, height=7.2)

#instruction on the change of output filenames
print("For c3.1a_offsetbaselined data, add letter-a after c3.2 to be c3.2a")
print("For c3.1b_pairwisebaselined data, add letter-b after c3.2 to be c3.2b")
```

Script code 4: PCA

```
#c4.1a PCA_total_ftir offset for salma's genotype paper

#imput file should be in csv format,
#typically, "xxxxxx_c2.1a_specpile_offset.csv" would be selected
#1st column should be the names of original spec files
#2nd col should be treatment either c3 or h3
#3rd col should be genotype, either chs, ima, or n61
#4th col should be identifier, which combines the 2nd and 3rd
#then followed by normalized abs values from 4000 to 400
#input data should have 4+3601=3605 columns.

#import necessary libraries
library(conflicted)
library(tidyverse)
```

```
library(psych)

#clean up the R's brain
rm(list=ls())

#obtain date information
today <- Sys.Date()
yr <- substr(today, 3,4)
mo <- substr(today, 6,7)
day <- substr(today, 9,10)
today2 <- paste(yr, mo, day, sep="")

#obtain the directory info for R script location
#ScriptPath <- getwd()

#obtain desktop folder information for a windows user
#you must change the string below within "xxx"according to your computer
username <- "akash"

#prepare output folder and its path
DesktopPath <- paste("C:/users/",username,"/desktop/", sep="")
setwd(DesktopPath)
if(!dir.exists(paste(today2, "_pca/", sep=""))){
   dir.create(paste(today2, "_pca/", sep=""), recursive=T)
}
OutputPath <- paste(DesktopPath, today2, "_pca/", sep="")

#invoke a file-opening window, specify the input file,
   #the input data should be in .csv format,
   #and has to be baseline-corrected and normalized
#obtain the filename
print("Please specify xxxxxx_c2.1a_specpile_offsetbaselined.csv")
inputfile <- file.choose()
filename <- basename(inputfile)
rawspecs <- read.csv(inputfile,
                                header = T)
filename

#extract values used for calculation to a new df specmatrix
#columns with zero values (wn4000, wn2600, wn2000, and wn400) should be removed
   #in "rawspecs", wn400 corresponds to col3605, thus the sum becomes 4005
   #thus, wn2000, wn2600, and wn4000 correspond to col
specmatrix <- dplyr::select(rawspecs, -(3605:3605))      #wn400
specmatrix <- dplyr::select(specmatrix, -(1:5)) #id and wn4000

#separate "identifier" column (category info)
id_2 <- dplyr::select(rawspecs, c(1,4))

#further trimming of values in the range of wn3600-4000
specmatrix <- dplyr::select(specmatrix, -(1:399))

#perform pca analysis using prcomp
#prcomp is standard but one of the oldest
pc = prcomp(specmatrix, scale =T)
```

```
#using the new 'principal()' in psych package
#pc <- psych::principal(specmatrix, nfactors=3601,
#                        rotate='none')

#display the summary
summary(pc)

#preparation of score data output
pc1_score <- pc$x[,1]
pc2_score <- pc$x[,2]
scoreonly <- as.data.frame(pc$x)
score <- cbind(id_2, scoreonly)

#export the score data as csv into the OutputPath
#data is PC1-PC2 score
setwd(OutputPath)
filename_PC12_score <- paste(today2,"_c4.1a_PCA_total_score_offset.csv", sep="")
write.csv(score,
          filename_PC12_score, row.names=FALSE)

#export rotation data as csv file
rotationonly <- as.data.frame(pc$rotation)
filename_pca_rotation <- paste(today2, "_c4.1a_pca_total_rotation_offset.csv", sep="")
write.csv(rotationonly, filename_pca_rotation, row.names=FALSE)

#export sdev data as csv file
sdevonly <- as.data.frame(pc$sdev)
filename_pca_sdev <- paste(today2, "_c4.1a_pca_total_sdev_offset.csv", sep="")
write.csv(sdevonly, filename_pca_sdev, row.names=FALSE)

#calculate the loadings, and export as csv file
loadingdata <- sweep(pc$rotation, MARGIN=2, pc$sdev, FUN="*")
filename_pca_loading <- paste(today2, "_c4.1a_pca_total_loading_offset.csv", sep="")
write.csv(loadingdata,
          filename_pca_loading, row.names=FALSE)

#draw the pc1_pc2 scoreplot
#size of the dots in the plot can be changed
   #by modifying the location of "geom_point(size=)
#color info can be seen in the following website
#http://sape.inf.usi.ch/quick-reference/ggplot2/colour
#point shape can be seen in the following website
#http://www.sthda.com/english/wiki/ggplot2-point-shapes
dev.new()
pca_total_scoreplot <- ggplot(score, aes(x = PC1,y = PC2,
        shape = identifier, color = identifier)) +
   geom_point(size=1) +
   scale_color_manual(values=c("deepskyblue","dodgerblue","dodgerblue4",
                               "salmon","salmon3","orangered")) +
   scale_shape_manual(values=c(1,2,4,1,2,4))+
#   scale_size_manual(values=c(10,1,1,10,1,1,10,1,1,10,1,1,10,1,1))+
   theme_bw()
print(pca_total_scoreplot)
```

```
#save the plot as png format
#you can change to .jpeg, .tiff, etc
#unit is in inch
filename_pca_total_scoreplot <- paste(today2, "_c4.1a_PCA_total_scoreplot_offset.png", sep="")
ggsave(file = filename_pca_total_scoreplot,
       plot = pca_total_scoreplot, dpi=600,
       width=7.2, height=4.8)

#Section 2: contribution check
#extract contribution data
contribution_total <- as.data.frame(t(summary(pc)$importance))
names(contribution_total) <-
c("standard_deviation","proportion_of_variance","cumulative_proportion")
contri_total_rownames <- as.data.frame(rownames(contribution_total))
names(contri_total_rownames) <- "PC"
contribution_total <- cbind(contribution_total, contri_total_rownames)

#save the contribution data as csv file
filename_pca_total_contribution <- paste(today2, "_c4.1a_pca_total_contribution_offset.csv",
sep="")
write.csv(contribution_total,
          filename_pca_total_contribution, row.names=FALSE)

#extract top 9 from the contribution data, and save as a png file
top9_contribution_total <- dplyr::slice(contribution_total, 1:9)

contribution_total_plot <- ggplot(top9_contribution_total, aes(x = PC, y =
proportion_of_variance)) +
  geom_bar(stat="identity", fill="forestgreen") +
  theme_bw()
print(contribution_total_plot)

#save the contribution graph
filename_pca_contribution_total_plot <- paste(today2,
"_c4.1a_PCA_contribution_total_offset.png", sep="")
ggsave(file = filename_pca_contribution_total_plot,
       plot = contribution_total_plot, dpi=100,
       width=7.2, height=4.8)

#Section 3: Loading plot for all the 6 groups
#prepare data for loading plot
pc_loading <- data.frame(t(cor(pc$x,specmatrix)))
pc_score <- data.frame(pc$x)

#draw the 2D-loading plot using ggplot
#color info can be seen in the following website
#http://sape.inf.usi.ch/quick-reference/ggplot2/colour

g0 <- ggplot()
g0 <- g0 + geom_segment(data=pc_loading,
                        aes(x=0,y=0,xend=(PC1*1),yend=(PC2*1)),
                        colour=rainbow(3200),alpha=0.2,size=0.5)
g0 <- g0 + xlab("PC1")
```

```
g0 <- g0 + ylab("PC2")
g0 <- g0 + theme_bw()
print(g0)

filename_pca_loading2d_plot <- paste(today2, "_c4.1a_PCA_loading2d_plot_offset.png", sep="")
ggsave(file = filename_pca_loading2d_plot,
       plot = g0, dpi=300, width=6.0, height=6.0)

#draw the 1D-loading barplot
#prepare x-axis data
wn_x_axis <- as.data.frame(seq(3600,401, by=-1))
names(wn_x_axis) <- c("wn")
pc_loading2 <- cbind(wn_x_axis, pc_loading)

#draw the PC1 loading barplot
g1 <- ggplot(data=pc_loading2,
             aes(x=wn, y=PC1))
g1 <- g1 + geom_bar(stat="identity", col=rainbow(3200))
g1 <- g1 + theme_bw()
print(g1)

filename_pca_PC1_loadingplot <- paste(today2, "_c4.1a_PCA_PC1_loadingplot_offset.png",
sep="")
ggsave(file = filename_pca_PC1_loadingplot,
       plot = g1, dpi=300, width=6.0, height=1.5)

#draw the PC2 loading barplot
g2 <- ggplot(data=pc_loading2,
             aes(x=wn, y=PC2))
g2 <- g2 + geom_bar(stat="identity", col=rainbow(3200))
g2 <- g2 + theme_bw()
print(g2)

filename_pca_PC2_loadingplot <- paste(today2, "_c4.1a_PCA_PC2_loadingplot_offset.png",
sep="")
ggsave(file = filename_pca_PC2_loadingplot,
       plot = g2, dpi=300, width=6.0, height=1.5)
```

Script code 5: FTIR marker boxplot

```
#c5.1_ftir marker boxplot
  #for salma's genotype paper
  #the 6 markers from the 1st paper are applied to the genotype data

#clear the brain
rm(list=ls())

#library to register
#ggplot2 and dplyr are in tidyverse
library(conflicted)
library(tidyverse)
library(MASS)
library(klaR)
library(caret)
```

```
library(ggpubr)

#obtain date information
today <- Sys.Date()
yr <- substr(today, 3,4)
mo <- substr(today, 6,7)
day <- substr(today, 9,10)
today2 <- paste(yr, mo, day, sep="")

#obtain desktop folder information for a windows user
#you must change the string below within "xxx"according to your computer
username <- "akash"

#prepare output folder and its path
DesktopPath <- paste("C:/users/",username,"/desktop/", sep="")
setwd(DesktopPath)
if(!dir.exists(paste(today2, "_MarkerBoxplot/", sep=""))){
   dir.create(paste(today2, "_MarkerBoxplot/", sep=""), recursive=T)
}
OutputPath <- paste(DesktopPath, today2, "_MarkerBoxplot/", sep="")

#redirect working directory and import the compiled ftir csv data
setwd("d:/1_DataFolder/Intel/i04_Informatics_Statistics/i04b_R/trainingdata/ftir_testdata/salma
_testdata/paper3/")
print("Please specify xxxxxx_specpile_offsetbaselined.csv")
specpile1 <- file.choose()
specpile2 <- read.csv(specpile1,
                      header = T)

#change the variable types of "condition" and "genotype"
#to factor format
specpile2$condition <- factor(specpile2$condition)
specpile2$genotype <- factor(specpile2$genotype)
specpile2$identifier <- factor(specpile2$identifier)

#import the "a6_anchorinfo.csv" file
   #integer of "1" should be input at col 7 and 8
print("Please specify anchor_info.csv")
specanchor1 <- file.choose()
specanchor2 <- read.csv(specanchor1,
                        header = T)

#obtain info on number of target markers
n_specanchor2 <- nrow(specanchor2)

#[section 0]: separation of each spec pair

#extract values used for calculation to a new df specmatrix
#in "specpile2", wn400 corresponds to col3605, thus the sum becomes 4005
#thus, wn4000 correspond to col

#split the data according to their identifiers
c3chs_specpile2 <- dplyr::filter(specpile2, identifier == 'c3chs')
c3ima_specpile2 <- dplyr::filter(specpile2, identifier == 'c3ima')
```

```
c3n61_specpile2 <- dplyr::filter(specpile2, identifier == 'c3n61')
h3chs_specpile2 <- dplyr::filter(specpile2, identifier == 'h3chs')
h3ima_specpile2 <- dplyr::filter(specpile2, identifier == 'h3ima')
h3n61_specpile2 <- dplyr::filter(specpile2, identifier == 'h3n61')

#combine the c3-h3 pairs
chs_spec <- rbind(c3chs_specpile2, h3chs_specpile2)
ima_spec <- rbind(c3ima_specpile2, h3ima_specpile2)
n61_spec <- rbind(c3n61_specpile2, h3n61_specpile2)

#make list of id-dataframe and id-name
list_spec_pair <- list(chs_spec, ima_spec, n61_spec)
list_chr_pair <- list("chs", "ima", "n61")

#make empty output dataframe
fmpileall <- data.frame(matrix(rep(NA, 8), nrow=1))[numeric(0),]
colnames(fmpileall) <- c("dataname", "identifier", "fm_numerator", "fm_denominator",
                         "fm", "abs_target", "abs_anchor1", "abs_anchor2")

#i-loop for different Fm markers
#for(i in 1:1){
for(i in 1:n_specanchor2){

  target_wn <- specanchor2[i,1]
  anchor1_wn <- specanchor2[i,2]
  anchor2_wn <- specanchor2[i,3]
  ylim_min <- specanchor2[i,7]
  ylim_max <- specanchor2[i,8]
  #specpile2 has 6 chr col before wn4000
  #thus, wn4000 corresponds to col7,
  #and wn400 corresponds to col3607
  #sum of these two becomes 4007
  col_target_wn <- 4005 - target_wn
  col_anchor1_wn <- 4005 - anchor1_wn
  col_anchor2_wn <- 4005 - anchor2_wn

  #prepare list for plots
  bplotlist <- list()

  #make empty output dataframe
  fmpilesub <- data.frame(matrix(rep(NA, 8), nrow=1))[numeric(0),]
  colnames(fmpilesub) <- c("dataname", "identifier", "fm_numerator", "fm_denominator",
                           "fm", "abs_target", "abs_anchor1", "abs_anchor2")

  #j-loop for different genotype
  for(j in 1:3){
  #for(j in 1:1){

    #calling working tissue data
    wspecpile <- list_spec_pair[j]
    wspecpile <- as.data.frame(wspecpile)
    wpairchr <- list_chr_pair[j]
    wpairchr <- unlist(wpairchr)
```

```
    #calculate fm value
    wspecpile2 <- dplyr::mutate(wspecpile, fm_numerator=(wspecpile[,col_target_wn] -
wspecpile[,col_anchor1_wn]))
    wspecpile2 <- dplyr::mutate(wspecpile2, fm_denominator=(wspecpile2[,col_anchor2_wn] -
wspecpile2[,col_anchor1_wn]))
    wspecpile2 <- dplyr::mutate(wspecpile2, fm=(fm_numerator/fm_denominator))
    wspecpile2 <- dplyr::mutate(wspecpile2, abs_target=wspecpile2[,col_target_wn])
    wspecpile2 <- dplyr::mutate(wspecpile2, abs_anchor1=wspecpile2[,col_anchor1_wn])
    wspecpile2 <- dplyr::mutate(wspecpile2, abs_anchor2=wspecpile2[,col_anchor2_wn])
    #colnames(specpile3)[(3608:3613)] <- c("fm_numerator", "fm_denominator", "fm"
    #                                      "abs_target",xxxxxx)
    wspecpile3 <- dplyr::select(wspecpile2, c(1, 4, 3606, 3607, 3608, 3609, 3610, 3611))

    fmpileall <- rbind(fmpileall, wspecpile3)
    fmpilesub <- rbind(fmpilesub, wspecpile3)

    #save the wspecpile3 as csv
    filename_wspecpile3 <- paste(today2, "_c5.1_fm", target_wn, "_", wpairchr, ".csv", sep="")
    setwd(OutputPath)
    atemp <- getwd()
    atemp
    write.csv(wspecpile3,
              filename_wspecpile3, row.names=FALSE)

    #make a boxplot
    #in the following, "x" should be the grouping variable,
    #usually in the category variable, such as condition
    #"y" should be numerical variable such as fm.
    #xlab("xxx") is for the label of figure
    #for color pallet, check the following
    # http://sape.inf.usi.ch/quick-reference/ggplot2/colour
    boxplot_title <- paste("fm", target_wn, "_", wpairchr, sep="")
    #dev.new()
    fm_boxplot <- ggplot(wspecpile3, aes(x = identifier, y = fm, fill=identifier)) +
      stat_boxplot(geom = "errorbar", width = 0.3)+
      geom_boxplot(outlier.size=1) +
      scale_fill_manual(values=c("deepskyblue", "salmon")) +
      #    geom_point(size=0.3, color='lightgray', alpha=0.5) +
      xlab("Condition") +
      ylab(boxplot_title) +
      #if you change the range of y-axis, use the follow line
      #    ylim(-20, 20)+
      theme_bw()
    #print(fm_boxplot)

    #save the same fm_boxplot as png file
    filename_fm_boxplot <- paste(today2, "_c6.1a_", boxplot_title, "_boxplot.png", sep="")
    setwd(OutputPath)
    atemp <- getwd()
    atemp
    ggsave(file = filename_fm_boxplot,
           plot = fm_boxplot, dpi = 100,
           width = 2.4, height = 2.4)
```

```
   bplotlist[[j]]<- fm_boxplot

   #end of j-loop
}

#k-loop for assembling 9 boxplots in 3x3 format in 1 figure
for (k in 1:9){
   allplot <- ggarrange(plotlist=c(bplotlist[1],
                                   bplotlist[2], bplotlist[3],
                                   bplotlist[4], bplotlist[5],
                                   bplotlist[6], bplotlist[7],
                                   bplotlist[8], bplotlist[9]),
            nrow=3, ncol=3, align="hv")

   #    print allplot
   setwd(OutputPath)
   tempa <- getwd()
   tempa
   filename_allplot <- paste(today2, "_c5.1_fm", target_wn,
                                        "_9boxplot.png", sep="")
   ggsave(file = filename_allplot,
             plot = allplot, dpi=100,
             width=14.4, height=7.2)

   #draw 3 subsets in one horizontal plot
   boxplot_title <- paste("fm", target_wn, sep="")
   fmpilesub2 <- transform(fmpilesub, identifier=factor(identifier,
      levels=c("c3chs", "h3chs", "c3ima","h3ima", "c3n61", "h3n61")))
   #dev.new()
   fm_horizonplot <- ggplot(fmpilesub2, aes(x = identifier, y = fm, fill=identifier)) +
      stat_boxplot(geom = "errorbar", width = 0.3)+
      geom_boxplot(outlier.size=1) +
      scale_fill_manual(values=c("deepskyblue", "salmon",
                                          "dodgerblue", "salmon2", "dodgerblue4", "salmon4")) +
      #    geom_point(size=0.3, color='lightgray', alpha=0.5) +
      xlab("Condition") +
      ylab(boxplot_title) +
      #if you change the range of y-axis, use the follow line
         ylim(ylim_min, ylim_max)+
      theme_bw()
   #print(fm_horizonplot)

   #save the same fm_boxplot as png file
   filename_fm_horizonplot <- paste(today2, "_c5.1_", boxplot_title, "_HorizonPlot.png", sep="")
   setwd(OutputPath)
   atemp <- getwd()
   atemp
   ggsave(file = filename_fm_horizonplot,
             plot = fm_horizonplot, dpi = 100,
             width = 7.2, height = 4.8)

   #end of k-loop
}
```

```
}
```

Script code 6: t-test for FTIR marker

```
#c5.2_t-test for ftir marker
   #for salma's genotype paper
   #results from c5.1_ftir_marker_boxplot are used

#clear the brain
rm(list=ls())

#library to register
#ggplot2 and dplyr are in tidyverse
library(conflicted)
library(tidyverse)
library(MASS)
library(klaR)
library(caret)
library(ggpubr)

#obtain date information
today <- Sys.Date()
yr <- substr(today, 3,4)
mo <- substr(today, 6,7)
day <- substr(today, 9,10)
today2 <- paste(yr, mo, day, sep="")

#obtain desktop folder information for a windows user
#you must change the string below within "xxx"according to your computer
username <- "akash"

#prepare output folder and its path
DesktopPath <- paste("C:/users/",username,"/desktop/", sep="")
setwd(DesktopPath)
if(!dir.exists(paste(today2, "_t-test/", sep=""))){
   dir.create(paste(today2, "_t-test/", sep=""), recursive=T)
}
OutputPath <- paste(DesktopPath, today2, "_t-test/", sep="")

#redirect working directory and import the compiled ftir csv data
setwd("d:/1_DataFolder/Intel/i04_Informatics_Statistics/i04b_R/trainingdata/ftir_testdata/salma
_testdata/paper3/ttest/")
print("Please specify xxxxxx_c5.1_fmxxx_genotype.csv")
df1 <- file.choose()
df2 <- read.csv(df1, header = T)

#perform t-test
t.test(fm ~ identifier, data=df2)
```

Script code 7: LDA

```
#c6.1 lda 2D genotype with offset baseline(400-4000) spec
   #linear discriminant analysis of ftir spectra
   #train-test sets were not prepared, and all data is used for modeling.
```

```
   #calculation using c3-h3 data in three genotypes

#this is a version for single baseline data

#clear the brain
rm(list=ls())

#library to register
#ggplot2 and dplyr are in tidyverse
library(conflicted)
library(tidyverse)
library(MASS)
library(klaR)
library(caret)
library(psych)
library(maptools)
library(ggrepel)

#obtain date information
today <- Sys.Date()
yr <- substr(today, 3,4)
mo <- substr(today, 6,7)
day <- substr(today, 9,10)
today2 <- paste(yr, mo, day, sep="")

# !!!    system check required, 1 out of 2
#obtain desktop folder information for a windows user
#you must change the string below within "xxx" according to your computer
username <- "akash"

#prepare output folder and its path
DesktopPath <- paste("C:/users/",username,"/desktop/", sep="")
setwd(DesktopPath)
if(!dir.exists(paste(today2, "_lda2d/", sep=""))){
   dir.create(paste(today2, "_lda2d/", sep=""), recursive=T)
}
OutputPath <- paste(DesktopPath, today2, "_lda2d/", sep="")

# !!!    system check required, 2 out of 2
#prepare input data folder
setwd("d:/1_DataFolder/Intel/i04_Informatics_Statistics/i04b_R/trainingdata/ftir_testdata/salma
_testdata/paper3/")
tempa <- getwd()
tempa

#invoke a file-opening window, specify the input file,
#the input data should be in .csv format,
   #and has to be baseline-corrected and normalized
#obtain the filename
print("Please specify xxxxxx_c2.1a_specpile_offsetbaselined.csv")
spec1 <- file.choose()
specpile <- read.csv(spec1,
                     header = T)
```

```
#extract values used for calculation to a new df specmatrix
#columns with zero values (wn400, 4000) should be removed
#moreover, data around noisy region of wn4000-3600 is also removed
#in "specpile", wn400 corresponds to col3605, thus the sum becomes 4005
#likewise, wn4000 correspond to col5
specpile2 <- dplyr::select(specpile, -(3605:3605))      #wn400
#specpile2 <- dplyr::select(specpile2, -(2007:2007))      #wn2000
#specpile2 <- dplyr::select(specpile2, -(1407:1407))      #wn2600
specpile2 <- dplyr::select(specpile2, -(5:405))      #wn4000-3600

#separate the identifier column (category info)
id_1 <- dplyr::select(specpile2, (1:4))

#extract values used for lda
specmatrix <- dplyr::select(specpile2, -(1:3))

#set the seednumber for randomness
set.seed(101)

#perform linear discriminant analysis
lda_specmatrix <- lda(identifier ~ ., specmatrix)

#calculate results
#1st, transform them to the values
#then one dimensional histograms
#"mar" is the margin of bottom, left, top, right
lda_specmatrix_results <- predict(lda_specmatrix)

#convert the $x score data to dataframe
ld_score <- data.frame(identifier=id_1[,4], lda=lda_specmatrix_results$x)

#draw 2D scatter plot using LD1-LD2 plain
dev.new()
g1 <- ggplot(ld_score, aes(x=lda.LD1, y=lda.LD2, colour=identifier, shape=identifier))+
   geom_point()+
   scale_color_manual(values=c("deepskyblue","dodgerblue","dodgerblue4",
                               "salmon","salmon3","orangered")) +
   scale_shape_manual(values=c(1,2,4,1,2,4))+
   theme_bw()
print(g1)

#save the plot as png format
setwd(OutputPath)
atemp <- getwd()
atemp
filename_lda2d <- paste(today2, "_c6.1_LDA2D.png", sep="")
ggsave(file = filename_lda2d,
          plot = g1, dpi=600,
          width=4.2, height=3.2)

#save the score data as csv file
ld_scoreonly <- dplyr::select(ld_score, -(1:1))
ld_score2 <- cbind(id_1, ld_scoreonly)
filename_ld_score2 <- paste(today2, "_c6.1_lda_score.csv", sep="")
```

```
setwd(OutputPath)
atemp <- getwd()
atemp
write.csv(ld_score2,
      filename_ld_score2, row.names=FALSE)

#extract LDA scalingdata
scaling1 <- lda_specmatrix$scaling

#transform LDA scaling to dataframe
#add wavenumber info
scaling2 <- as.data.frame(t(scaling1))
wnlist <- seq(from=3599, to=401, by=-1)
#    wnlist_frag1 <- seq(from=3599, to=2601, by=-1)
#    wnlist_frag2 <- seq(from=2599, to=2001, by=-1)
#    wnlist_frag3 <- seq(from=1999, to=401, by=-1)
#    wnlist <- c(wnlist_frag1, wnlist_frag2, wnlist_frag3)
wnlist2 <- as.data.frame(t(wnlist))

colnames(scaling2) <- wnlist
colnames(wnlist2) <- wnlist

scaling3 <- rbind(wnlist2, scaling2)
scaling4 <- as.data.frame(t(scaling3))
names(scaling4)[1] <- "wavenumber"

#change the wavenumber in ascending order, and save it as csv
scaling5 <- arrange(scaling4, wavenumber)
filename_scaling5 <- paste(today2, "_c6.1_lda_scaling.csv", sep="")
write.csv(scaling5,
          filename_scaling5, row.names=FALSE)

#plot LD1 scaling, scatter plot version
dev.new()
lda_scaling_LD1_scatterplot <- ggplot(scaling5, aes(x = wavenumber, y = LD1)) +
      geom_point(size=0.5) +
      theme_bw()
   print(lda_scaling_LD1_scatterplot)

   filename_lda_scaling_LD1_scatterplot <- paste(today2, "_c6.1_Scaling_LD1_ScatterPlot.png",
sep="")
   ggsave(file = filename_lda_scaling_LD1_scatterplot,
             plot = lda_scaling_LD1_scatterplot, dpi = 300,
             width = 7.2, height = 2.4)

#plot LD1 scaling, rainbow line plot version
   dev.new()
   gscale_ld1_rb <- ggplot(data=scaling5,
                           aes(x = wavenumber, y = LD1))
   gscale_ld1_rb <- gscale_ld1_rb + geom_bar(stat="identity", col=rainbow(3199))
   gscale_ld1_rb <- gscale_ld1_rb + theme_bw()
   print(gscale_ld1_rb)

   filename_gscale_ld1_rb <- paste(today2, "_c6.1_Scaling_LD1_RainbowLinePlot.png", sep="")
```

```
   ggsave(file = filename_gscale_ld1_rb,
            plot = gscale_ld1_rb, dpi = 300,
            width = 7.2, height = 2.4)

#plot LD2 scaling, scatter plot version
dev.new()
lda_scaling_LD2_scatterplot <- ggplot(scaling5, aes(x = wavenumber, y = LD2)) +
      geom_point(size=0.5) +
      theme_bw()
   print(lda_scaling_LD2_scatterplot)

   filename_lda_scaling_LD2_scatterplot <- paste(today2, "_c6.1_Scaling_LD2_ScatterPlot.png",
sep="")
   ggsave(file = filename_lda_scaling_LD2_scatterplot,
            plot = lda_scaling_LD2_scatterplot, dpi = 300,
            width = 7.2, height = 2.4)

#plot LD2 scaling, rainbow line plot version
   dev.new()
   gscale_ld2_rb <- ggplot(data=scaling5,
                                 aes(x = wavenumber, y = LD2))
   gscale_ld2_rb <- gscale_ld2_rb + geom_bar(stat="identity", col=rainbow(3199))
   gscale_ld2_rb <- gscale_ld2_rb + theme_bw()
   print(gscale_ld2_rb)

   filename_gscale_ld2_rb <- paste(today2, "_c6.1_Scaling_LD2_RainbowLinePlot.png", sep="")
   ggsave(file = filename_gscale_ld2_rb,
            plot = gscale_ld2_rb, dpi = 300,
            width = 7.2, height = 2.4)

   #plot LD1-LD2 scaling 2D plot
   dev.new()
   g2d <- ggplot()
   g2d <- g2d + geom_point(data=scaling5,
                           aes(x = LD1, y = LD2),
                           colour=rainbow(3199),
                           alpha=0.8, size=2)
   g2d <- g2d + labs()
   g2d <- g2d + theme_bw()
   print(g2d)

#save the plot as png format
   filename_g2d <- paste(today2, "_c6.1_Scaling_2D_LD12_ScatterPlot.png", sep="")
   ggsave(file = filename_g2d,
            plot = g2d, dpi=600,
            width=7.2, height=4.8)
```

Script code 8: Drawing magnified spectra in the vicinity of Fm marker

```
#drawspec_markervicinity for salma's genotype paper
   #for drawing spectrum in the vicinity of ftir-marker
   #this is for Salma's data on c3-h3 chamber comparison.

#input file is "specmean.csv"
```

```
#clear the brain
rm(list=ls())

#library to register
#ggplot2 and dplyr are in tidyverse
library(conflicted)
library(tidyverse)
library(MASS)
library(klaR)
library(caret)
library(ggpubr)

#obtain date information
today <- Sys.Date()
yr <- substr(today, 3,4)
mo <- substr(today, 6,7)
day <- substr(today, 9,10)
today2 <- paste(yr, mo, day, sep="")

# !!!    system check required, 1 out of 2
#obtain desktop folder information for a windows user
#you must change the string below within "xxx" according to your computer
username <- "akash"

#prepare output folder and its path
DesktopPath <- paste("C:/users/",username,"/desktop/", sep="")
setwd(DesktopPath)
if(!dir.exists(paste(today2, "_drawspec_markervicinity/", sep=""))){
   dir.create(paste(today2, "_drawspec_markervicinity/", sep=""), recursive=T)
}
OutputPath <- paste(DesktopPath, today2, "_drawspec_markervicinity/", sep="")

# !!!    system check required, 2 out of 2
#prepare input data folder
setwd("d:/1_DataFolder/Intel/i04_Informatics_Statistics/i04b_R/trainingdata/ftir_testdata/salma
_testdata/paper3")
a <- getwd()
a

#import a pair of spec ftir data
   #it should be 3602 cols, 1st col is an identifier, then wn4000-400
print("Please specify xxxxxx_c3.1a_specmean.csv")
specpile1 <- file.choose()
sixspec <- read.csv(specpile1,
                                header = T)

#import the 2nd, "xxxxxx_newmarker_info.csv" file
print("Please specify xxxxxx_newmarker_info.csv")
anchor_info <- file.choose()
newmarker_info2 <- read.csv(anchor_info,
                                   header = T)

#arrange the spectra
```

```
sixspec2 <- dplyr::select(sixspec, -(c(1:1)))
colnames(sixspec2) <- seq(from=4000, to=400, by=-1)

wnlist1 <- as.data.frame(t(seq(from=4000, to=400, by=-1)))
colnames(wnlist1) <- seq(from=4000, to=400, by=-1)
sixspec3 <- rbind(wnlist1, sixspec2)

longspec3 <- as.data.frame(t(sixspec3))
colnames(longspec3) <- c("wn", "c3chs", "c3ima", "c3n61",
                         "h3chs", "h3ima", "h3n61")

#draw the entire spec
#color info can be seen in the following website
#http://sape.inf.usi.ch/quick-reference/ggplot2/colour

dev.new()
glongspec3 <- ggplot(longspec3) +
  theme_bw()+
  geom_line(aes(x=wn, y=c3chs),
            colour="deepskyblue", size=0.3)+
  geom_line(aes(x=wn, y=c3ima),
            colour="springgreen3", size=0.3)+
  geom_line(aes(x=wn, y=c3n61),
            colour="dodgerblue4", size=0.3)+
  geom_line(aes(x=wn, y=h3chs),
            colour="salmon", size=0.3)+
  geom_line(aes(x=wn, y=h3ima),
            colour="deeppink", size=0.3)+
  geom_line(aes(x=wn, y=h3n61),
            colour="orangered4", size=0.3)
  print(glongspec3)

#save the plot as png format(you can change to .jpeg, .tiff, etc)
setwd(OutputPath)
b <- getwd()
b
filename_glongspec3 <- paste(today2, "_c7.1_longspec3.png", sep="")
ggsave(file = filename_glongspec3,
       plot = glongspec3, dpi=100,
       width=3.6, height=1.2)

#set the target and anchor wavenumbers
nrow_newmarker_info2 <- nrow(newmarker_info2)

#prepare list for plots
plotrawtrace <- list()
plotlistnorm <- list()
plotlistnorm2 <- list()

#i-loop for drawing respective trace around target wavenumbers
  #in the parameters below, wn_anchor1 and 2 are
  #the ones with lower and higher absorbance (or valley and peak)

for (i in 1:nrow_newmarker_info2){
```

```
#for (i in 1:1){
   wn_target1 <- newmarker_info2[i,1]
   wn_target2 <- newmarker_info2[i,2]
   wn_target3 <- newmarker_info2[i,3]
   wn_target4 <- newmarker_info2[i,4]
   wn_anchor1 <- newmarker_info2[i,5]
   wn_anchor2 <- newmarker_info2[i,6]
   wn_Ledge <- newmarker_info2[i,7]
   wn_Hedge <- newmarker_info2[i,8]

   #obtaining row numbers for the target and its frame edges in longspec3
   row_target1 <- 4001 - wn_target1
   row_target2 <- 4001 - wn_target2
   row_target3 <- 4001 - wn_target3
   row_anchor1 <- 4001 - wn_anchor1
   row_anchor2 <- 4001 - wn_anchor2
   row_Ledge <- 4001 - wn_Ledge
   row_Hedge <- 4001 - wn_Hedge

   # [Option 1] raw trace without anchoring

   #slice the rows for the magnified frame
   specmag <- dplyr::slice(longspec3, row_Hedge:row_Ledge)

   #save the specmag data as csv
   setwd(OutputPath)
   b <- getwd()
   b
   filename_specmag <- paste(today2, "_c7.1_specraw_", wn_target1, "_", wn_target2, ".csv",
sep="")
   write.csv(specmag, filename_specmag, row.names=FALSE)

   #draw the sliced region of the spec
   dev.new()
   plotspecmag <- ggplot(specmag) +
      theme_bw()+
      geom_line(aes(x=wn, y=c3chs),
                colour="deepskyblue", size=0.3)+
      geom_line(aes(x=wn, y=c3ima),
                colour="springgreen3", size=0.3)+
      geom_line(aes(x=wn, y=c3n61),
                colour="dodgerblue4", size=0.3)+
      geom_line(aes(x=wn, y=h3chs),
                colour="salmon", size=0.3)+
      geom_line(aes(x=wn, y=h3ima),
                colour="deeppink", size=0.3)+
      geom_line(aes(x=wn, y=h3n61),
                colour="orangered4", size=0.3)+
      geom_vline(xintercept=wn_target1,
                 colour="orange",size=0.3)+
      geom_vline(xintercept=wn_target2,
                 colour="green",size=0.3)+
      geom_vline(xintercept=wn_target3,
                 colour="blue",size=0.3)+
```

```
    geom_vline(xintercept=wn_target4,
                colour="magenta",size=0.3)+
    #       geom_vline(xintercept=wn_anchor1,
#                   colour="magenta", size=0.3)+
#     geom_vline(xintercept=wn_anchor2,
#                   colour="blue", size=0.3)+
    theme(axis.text.x=element_text(angle=45, hjust=1))
  print(plotspecmag)
  plotrawtrace[[i]] <- plotspecmag

  #save the plot as png format
  setwd(OutputPath)
  b <- getwd()
  b
  filename_plotspecmag <- paste(today2, "_c7.1_specraw_",wn_target1, "_", wn_target2, ".png",
sep="")
  ggsave(file = filename_plotspecmag,
         plot = plotspecmag, dpi=300,
         width=3.6, height=3.6)

  # [Option 2] normalized trace using 2 anchors

  #extract the values for anchors 1 and 2
  abs_c3chs_anchor1 <- longspec3[row_anchor1,2]
  abs_c3chs_anchor2 <- longspec3[row_anchor2,2]
  abs_c3ima_anchor1 <- longspec3[row_anchor1,3]
  abs_c3ima_anchor2 <- longspec3[row_anchor2,3]
  abs_c3n61_anchor1 <- longspec3[row_anchor1,4]
  abs_c3n61_anchor2 <- longspec3[row_anchor2,4]
  abs_h3chs_anchor1 <- longspec3[row_anchor1,5]
  abs_h3chs_anchor2 <- longspec3[row_anchor2,5]
  abs_h3ima_anchor1 <- longspec3[row_anchor1,6]
  abs_h3ima_anchor2 <- longspec3[row_anchor2,6]
  abs_h3n61_anchor1 <- longspec3[row_anchor1,7]
  abs_h3n61_anchor2 <- longspec3[row_anchor2,7]

  #calculate the normalized absorbance (nabs) using Fm formula
  #the formula is (A_target - A_anchor1)/(A_anchor2 - A_anchor1)

  specmag2 <- mutate(specmag, nabs_c3chs=(c3chs-abs_c3chs_anchor1)/(abs_c3chs_anchor2-
abs_c3chs_anchor1))
  specmag2 <- mutate(specmag2, nabs_c3ima=(c3ima-abs_c3ima_anchor1)/(abs_c3ima_anchor2-
abs_c3ima_anchor1))
  specmag2 <- mutate(specmag2, nabs_c3n61=(c3n61-abs_c3n61_anchor1)/(abs_c3n61_anchor2-
abs_c3n61_anchor1))
  specmag2 <- mutate(specmag2, nabs_h3chs=(h3chs-abs_h3chs_anchor1)/(abs_h3chs_anchor2-
abs_h3chs_anchor1))
  specmag2 <- mutate(specmag2, nabs_h3ima=(h3ima-
abs_h3ima_anchor1)/(abs_h3ima_anchor2-abs_h3ima_anchor1))
  specmag2 <- mutate(specmag2, nabs_h3n61=(h3n61-
abs_h3n61_anchor1)/(abs_h3n61_anchor2-abs_h3n61_anchor1))

  #save the specmag2 data as csv
  setwd(OutputPath)
```

```
   b <- getwd()
   b
   filename_specmag2 <- paste(today2, "_c7.1_specmag2_", wn_target1, "_", wn_target2, ".csv",
sep="")
   write.csv(specmag2, filename_specmag2, row.names=FALSE)

   #draw the normalized spec
   dev.new()
   plotspecmag2n <- ggplot(specmag2) +
      theme_bw()+
      geom_line(aes(x=wn, y=nabs_c3chs),
                  colour="deepskyblue", size=0.3)+
      geom_line(aes(x=wn, y=nabs_c3ima),
                  colour="deepskyblue2", size=0.3)+
      geom_line(aes(x=wn, y=nabs_c3n61),
                  colour="deepskyblue4", size=0.3)+
      geom_line(aes(x=wn, y=nabs_h3chs),
                  colour="salmon", size=0.3)+
      geom_line(aes(x=wn, y=nabs_h3ima),
                  colour="salmon2", size=0.3)+
      geom_line(aes(x=wn, y=nabs_h3n61),
                  colour="salmon4", size=0.3)+
      geom_vline(xintercept=wn_target1,
                   colour="orange",size=0.3)+
      geom_vline(xintercept=wn_target2,
                   colour="green",size=0.3)+
      geom_vline(xintercept=wn_target3,
                   colour="blue",size=0.3)+
      geom_vline(xintercept=wn_target4,
                   colour="magenta",size=0.3)+
      theme(axis.text.x=element_text(angle=45, hjust=1))
   print(plotspecmag2n)
   plotlistnorm[[i]] <- plotspecmag2n

   #save the plot as png format
   setwd(OutputPath)
   b <- getwd()
   b
   filename_plotspecmag2n <- paste(today2, "_c7.1_specmag2n_",wn_target1, "_", wn_target2,
".png", sep="")
   ggsave(file = filename_plotspecmag2n,
            plot = plotspecmag2n, dpi=300,
            width=3.6, height=3.6)
   #this is the end of magnification option 2

} # this is an end of the i-loop
```