

# Wood preservatives utilizing low-value olive oil production by-products: Analysis

\*\*\*\*\*

*Michael Burnard, University of Primorska Muzejski trg 2, 6000 Koper, Slovenia; InnoRenew  
CoE, Livade 6, 6310 Izola, Slovenia*

*Matthew Schwarzkopf, University of Primorska Muzejski trg 2, 6000 Koper, Slovenia;  
InnoRenew CoE, Livade 6, 6310 Izola, Slovenia*

*Viacheslav Tverezovskiy, BioComposites Centre Bangor University, LL57 2UW, Bangor,  
Gwynedd, UK*

*Andreas Treu, Norwegian Institute of Bioeconomy Research Postboks 115, NO-1431 Ås,  
Norway*

*Miha Humar, Department of Wood Science and Technology, Biotechnical Faculty, University  
of Ljubljana, Rožina dolina, Cesta VIII/34, 1000 Ljubljana*

*Andreja Kutnar, University of Primorska Muzejski trg 2, 6000 Koper, Slovenia; InnoRenew  
CoE, Livade 6, 6310 Izola, Slovenia*

\*\*\*\*\*

*16 February 2018*

# Contents

<b>Introduction</b>	<b>3</b>
Set up the environment . . . . .	3
<b>Leaching</b>	<b>3</b>
Load data set . . . . .	3
Clean, organise, and modify the data. . . . .	4
Visualisation . . . . .	5
Mass changes as a percent of original mass . . . . .	5
Mass changes in grams . . . . .	7
Statistics and models . . . . .	8
Some summary data . . . . .	9
Significance tests . . . . .	10
<b>Fungi Results</b>	<b>16</b>
Read and prepare the data . . . . .	16
Comparisons . . . . .	20
Beech :: Inoculant PL . . . . .	21
Beech :: Inoculant TV . . . . .	24
Pine :: Inoculant GT . . . . .	27
Pine :: Inoculant PM . . . . .	30
Oil Extraction . . . . .	33
Fungal block test comparisons . . . . .	34
Beech . . . . .	34
Pine . . . . .	36
<b>Acknowledgements</b>	<b>39</b>
<b>Environment</b>	<b>40</b>

## Introduction

Pine and beech wood specimens were treated with three types of preservative treatments and exposed to leaching and mini block fungal decay tests. The preservative treatments were developed from by-products of olive oil production and were intended to:

- Reduce leaching of the preservative out of the wood
- Increase hydrophobicity
- Impart some level of antimicrobial performance

The four treatments were:

- Lampante Oil (LO), non-modified
- Direct Maleinization of LO (DM)
- Glycerolysis Maleinization of LO (GM)
- Water / untreated (Control)

## Set up the environment

Load these packages to perform the analysis and create the visualisations.

```
#basic packages  
library(tidyverse)  
library(scales)  
library(zoo)  
library(gridExtra)  
library(ggalt)
```

## Leaching

Leaching tests consisted of 4 wet-dry cycles. The mass and dimensions of the specimens were measured and then submersed in distilled water for 48 hours. After the wet period their dimensions were measured. They were then dried in a climate chamber at 40 C for 24 hours. After the drying period, the mass and dimensions were measured again. This procedure was completed for 4 cycles.

Within this analysis we will refer to the treatments as LO, DM, GM, and Control (water).

## Load data set

Load the most recent data, and remove the first column (just row numbers). Then take a look at its structure.

```
d <- read.csv("data/00TLeaching1MS-150417_v1_r0.csv", stringsAsFactors=FALSE)  
d <- d[complete.cases(d),] # only weights for now.  
glimpse(d)
```

```
## Observations: 400  
## Variables: 8  
## $ Specimen <chr> "P-Control-1", "P-Control-2", "P-Control-3", "P-Contr...  
## $ Date <chr> "11/10/16", "11/10/16", "11/10/16", "11/10/16", "11/1...  
## $ Time <chr> "13:00", "13:00", "13:00", "13:00", "13:00", "11:00",...  
## $ Length <dbl> 29.86, 29.82, 29.89, 29.94, 29.85, 29.83, 29.94, 30.0...  
## $ Width <dbl> 30.22, 30.09, 29.95, 30.07, 30.04, 30.05, 30.09, 30.1...
```

```
## $ Depth <dbl> 15.26, 15.32, 15.43, 15.31, 15.19, 15.20, 15.16, 15.1...
## $ Mass <dbl> 9.0280, 9.4440, 9.4250, 9.4950, 8.9940, 9.4320, 9.390...
## $ Initials <chr> "MS", "MS", "MS", "MS", "MS", "MS", "MS", "MS", "MS",...
```

## Clean, organise, and modify the data.

First, organise the data and add some extra data of interest.

Add a nicely named identifier column, *Cycle* that is equivalent to dates, and blocking variable.

```
d$Cycle <- ifelse(d$Date == "11/10/16", "Cycle-0",
  ifelse(d$Date == "11/15/16", "Cycle-1",
    ifelse(d$Date == "11/18/16", "Cycle-2",
      ifelse(d$Date == "11/21/16", "Cycle-3",
        ifelse(d$Date == "11/28/16", "Cycle-4", NA))))))
d$Block <- ifelse(is.na(d$Cycle), "B", "A")
d$Cycle <- ifelse(d$Date == "1/5/17", "Cycle-0",
  ifelse(d$Date == "1/8/17", "Cycle-1",
    ifelse(d$Date == "1/11/17", "Cycle-2",
      ifelse(d$Date == "1/14/17", "Cycle-3",
        ifelse(d$Date == "1/19/17", "Cycle-4", d$Cycle))))))
```

- *dffPrev* and *perDffPrev* are mass differences from the previous cycle.

```
d <- d %>% arrange(Block, Specimen) %>% group_by(Specimen) %>%
  mutate(dffPrev = Mass - lag(Mass, default=Mass[1]),
    perDffPrev = (Mass - lag(Mass, default=Mass[1]))/Mass[1])
```

Now, to calculate the weight difference from the original value for each specimen after each cycle.

- *origMass* is a column that contains the original mass for each specimen.

The `na.locf()` function is from the `zoo` package, and fills in the NA values with the value from the previous row.

- *totalDff* is the difference between the weight at the current cycle compared to the original mass
- *perDffOrig* is same value as a percent of the original mass

```
d$origMass <- ifelse(d$Cycle == "Cycle-0", d$Mass, NA)
d$origMass <- na.locf(d$origMass)

d <- d %>% group_by(Specimen) %>% mutate(totalDff = Mass - origMass,
  perDffOrig = ((Mass - origMass) / origMass))
```

A little clean up... Split the Specimen name column into separate values for modeling and summaries.

```
d <- d %>% separate(Specimen, c("Species", "Treatment", "Test", "SpecNo"),
  sep="-", remove=FALSE, fill="right", extra="warn")

d$SpecNo <- ifelse(is.na(d$SpecNo), d$Test, d$SpecNo)
d <- d[,-4]
```

Now the data is ready for visualisation and further analysis.

## Visualisation

First, we create a second data set with only the rows we need. Then, we do some summary statistics to help us annotate the charts we'll make.

```
d2 <- d %>% filter(!Cycle == "Cycle-0")
anno_text <- d2 %>% group_by(Cycle) %>% summarise(muPer=round(mean(perDffOrig),5),
                                                muTot=round(mean(totalDff),4))

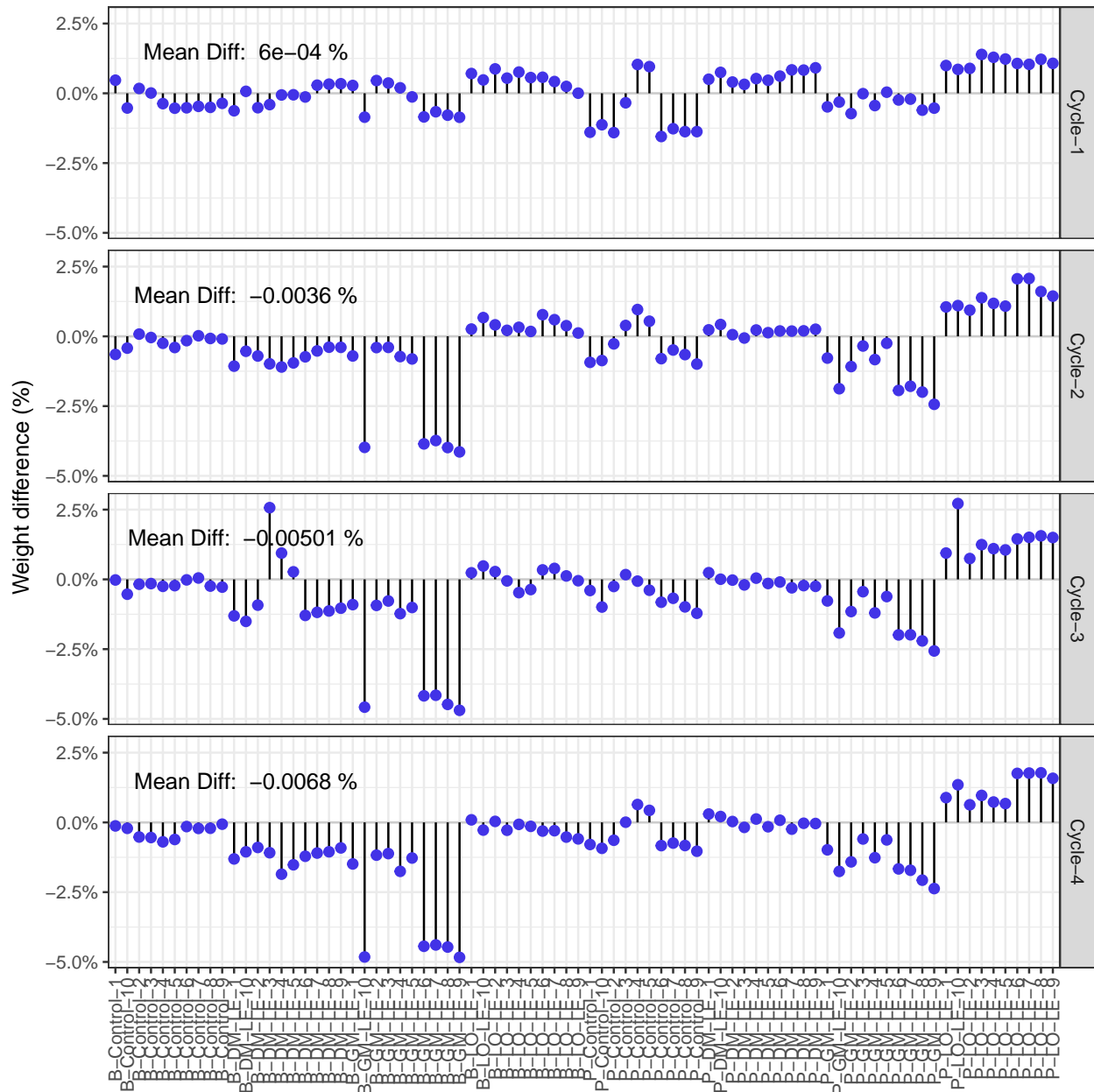
anno_text$muPer <- paste("Mean Diff: ", as.character(anno_text$muPer), "%")
anno_text$muTot <- paste("Mean Diff: ", as.character(anno_text$muTot), "g")
```

### Mass changes as a percent of original mass

```
ggplot(data=d2, aes(x=Specimen, y=perDffOrig)) + theme_bw() +
  geom_hline(yintercept=0, colour="#CCCCCC") +
  geom_lollipop(point.colour="#4233E7", point.size=2) +
  facet_grid(Cycle~.) +
  scale_y_continuous(labels=percent) +
  geom_text(data=anno_text, aes(x=12,y=.015, label=muPer)) +
  labs(y="Weight difference (%)", x=NULL,
       title="Percent weight difference after drying phase",
       subtitle="Difference = (Cycle # - Original) / Original") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1, vjust = 0.5))
```

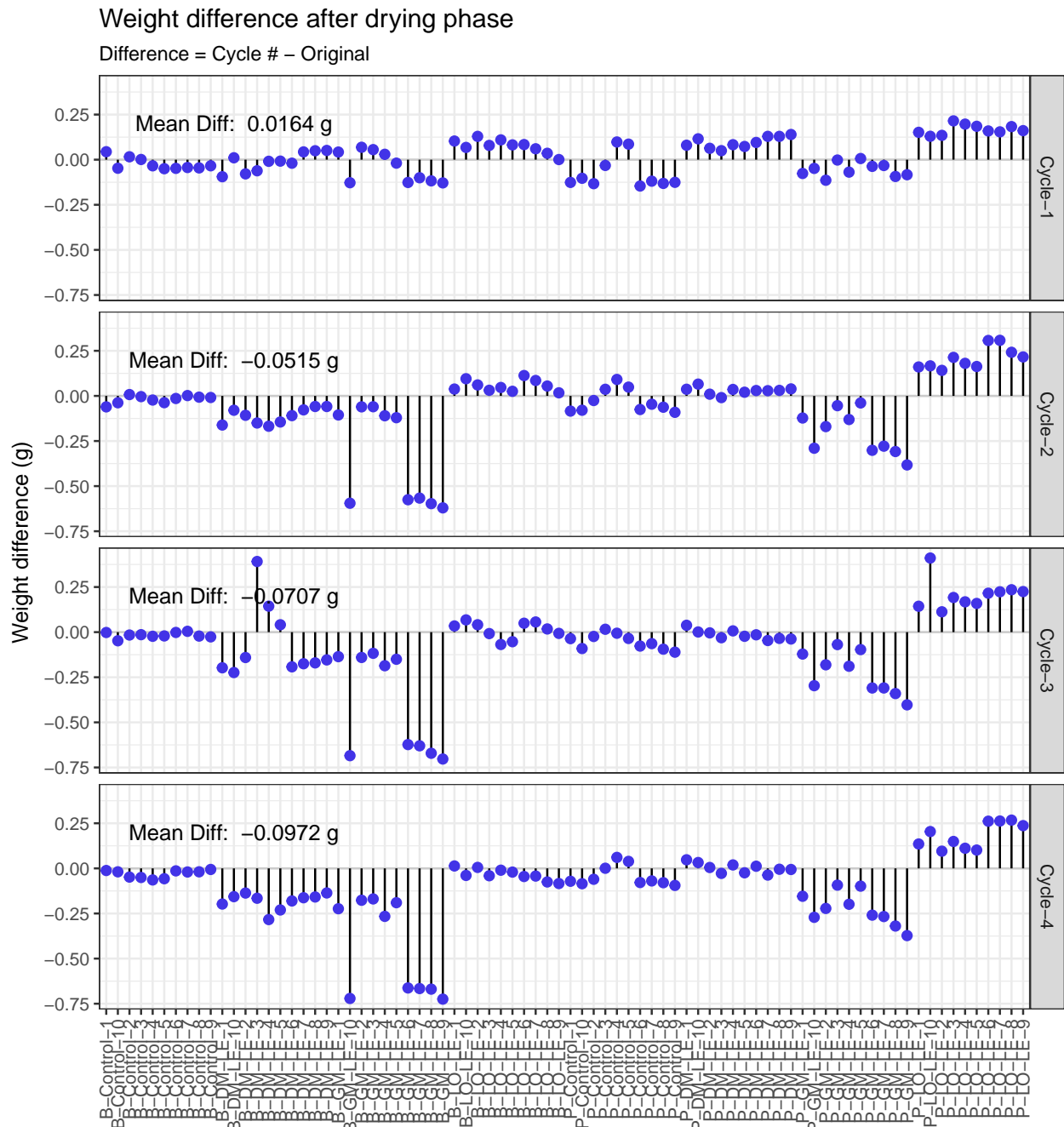
# Percent weight difference after drying phase

$$\text{Difference} = (\text{Cycle \#} - \text{Original}) / \text{Original}$$



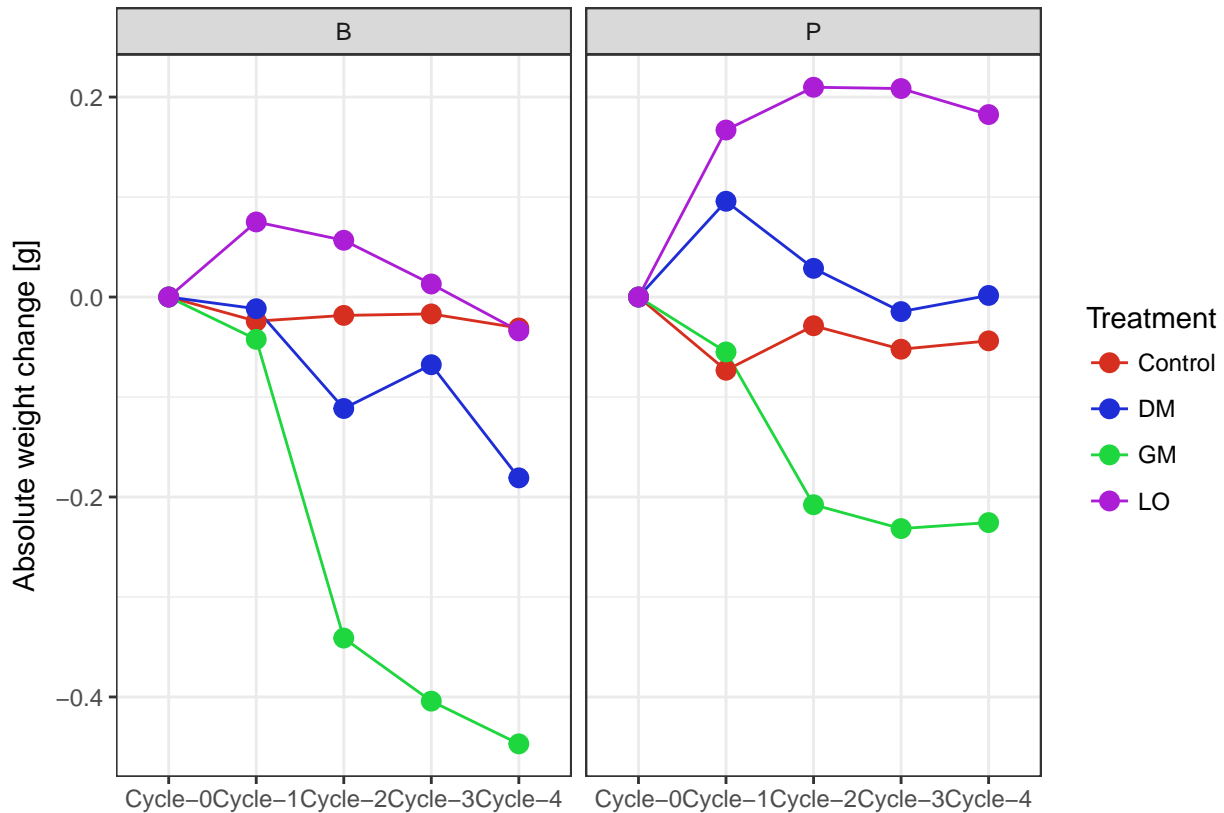
## Mass changes in grams

```
ggplot(d2, aes(x=Specimen, y=totalDff)) + theme_bw() +  
  geom_hline(yintercept=0, colour="#CCCCCC") +  
  geom_lollipop(point.colour="#4233E7", point.size=2) +  
  facet_grid(Cycle~.) +  
  geom_text(data=anno_text, aes(x=12,y=.2, label=muTot)) +  
  labs(y="Weight difference (g)", x=NULL,  
       title="Weight difference after drying phase",  
       subtitle="Difference = Cycle # - Original") +  
  theme(axis.text.x = element_text(angle = 90, hjust = 1, vjust = 0.5))
```



This is a convenient summary of weight change by species and treatment from cycle to cycle.

```
leach.sum <- d %>% group_by(Treatment,Species,Cycle) %>% summarise(meanDff = mean(totalDff))
ggplot(leach.sum, aes(x=Cycle, y=meanDff, colour=Treatment, group=Treatment)) + theme_bw() +
  geom_line(size=0.5) +
  geom_point(size = 3) +
  scale_colour_manual(values=c("#D62F1F", "#1E2DD6", "#1ED63D", "#AB1ED6")) +
  facet_grid(.~Species) +
  labs(y = "Absolute weight change [g]", x="")
```



## Statistics and models

First, lets get just the data we need and take a look at it. The data is for the final Cycle only, so the differences are the total difference from the original state to the final state.

```
ds <- d %>% filter(Cycle=="Cycle-4") %>% select(Specimen, Species, Treatment, Block, Mass,
  totalDff, SpecNo, perDffOrig, Cycle)
glimpse(ds)
```

```
## Observations: 80
## Variables: 9
## $ Specimen <chr> "B-Control-1", "B-Control-2", "B-Control-3", "B-Con...
## $ Species <chr> "B", "B", "B", "B", "B", "B", "B", "B", "B", "B", "...
## $ Treatment <chr> "Control", "Control", "Control", "Control", "Contro...
## $ Block <chr> "A", "A", "A", "A", "A", "A", "A", "A", "A", "A", "...
## $ Mass <dbl> 9.3592, 9.1668, 9.2486, 9.1419, 9.3197, 14.9069, 15...
## $ totalDff <dbl> -0.0118, -0.0482, -0.0504, -0.0641, -0.0573, -0.197...
## $ SpecNo <chr> "1", "2", "3", "4", "5", "1", "2", "3", "4", "5", "...
```



```
## $ perDffOrig <dbl> -1.259204e-03, -5.230602e-03, -5.419938e-03, -6.962...
## $ Cycle      <chr> "Cycle-4", "Cycle-4", "Cycle-4", "Cycle-4", "Cycle-...
```

### Some summary data

By treatment and species.

```
ds %>% group_by(Species, Treatment) %>% summarise(T.med=median(totalDff), T.sd=sd(totalDff),
                                                  P.med=median(perDffOrig), P.sd=sd(perDffOrig),
                                                  N.obs=n())
```

Species	Treatment	T.med	T.sd	P.med	P.sd	N.obs
B	Control	-0.01955	0.0214879	-0.0021379	0.0023144	10
B	DM	-0.16410	0.0458327	-0.0109228	0.0029470	10
B	GM	-0.46435	0.2570854	-0.0307204	0.0171695	10
B	LO	-0.03995	0.0316664	-0.0027807	0.0022110	10
P	Control	-0.07045	0.0559437	-0.0076551	0.0060216	10
P	DM	0.00005	0.0271260	0.0000023	0.0017371	10
P	GM	-0.24055	0.0912156	-0.0153969	0.0058751	10
P	LO	0.17655	0.0713799	0.0115770	0.0048330	10

## Significance tests

Assumptions were not met for parametric tests, so we use non-parametric alternatives. Separate by species (does not make sense to compare them this way) – there is no expectation they would behave similarly.

```
ds.b <- ds %>% filter(Species == "B")
ds.p <- ds %>% filter(Species == "P")
ds.b$Treatment <- factor(ds.b$Treatment)
ds.p$Treatment <- factor(ds.p$Treatment)
ds.b$Block <- factor(ds.b$Block)
ds.p$Block <- factor(ds.p$Block)
```

### Beech by Block

No difference detected.

```
wilcox.test(perDffOrig ~ Block, data=ds.b)
```

```
##
## Wilcoxon rank sum test
##
## data: perDffOrig by Block
## W = 213, p-value = 0.7381
## alternative hypothesis: true location shift is not equal to 0
```

### Beech by Block

No difference detected.

```
wilcox.test(perDffOrig ~ Block, data=ds.p)
```

```
##
## Wilcoxon rank sum test
##
## data: perDffOrig by Block
## W = 237, p-value = 0.3273
## alternative hypothesis: true location shift is not equal to 0
```

### Beech by Treatment

Treatment does seem to have an effect for beech specimens.

```
kruskal.test(perDffOrig ~ Treatment, data=ds.b)
```

```
##
## Kruskal-Wallis rank sum test
##
## data: perDffOrig by Treatment
## Kruskal-Wallis chi-squared = 31.259, df = 3, p-value = 7.499e-07
```

### Pine by treatment

Treatment does seem to have an effect for pine specimens.

```
kruskal.test(perDffOrig ~ Treatment, data=ds.p)
```

```
##
## Kruskal-Wallis rank sum test
##
## data: perDffOrig by Treatment
## Kruskal-Wallis chi-squared = 31.15, df = 3, p-value = 7.903e-07
```

No apparent differences between blocks as a whole. There seem to be clear differences between treatments within species.

Now we test for differences between treatments within species.

### Beech treatment comparisons

To determine if there were differences between treatments, the effect size, and confidence intervals for the effect size, we use Wilcoxon rank sum tests for each pairing. We save the results to a dataframe for further inspection. We repeat the procedure the pine leaching data, in the following section.

```
bl.df <- data_frame(Treat1=as.character("treat1"), Treat2=as.character("treat2"),
                    Loc.shift = 0, CI.lwr = 0, CI.upr = 0, p.value = 0)
b.cl <- ds.b %>% filter(Treatment %in% c("Control", "LO"))
b.cg <- ds.b %>% filter(Treatment %in% c("Control", "GM"))
b.cd <- ds.b %>% filter(Treatment %in% c("Control", "DM"))
b.lg <- ds.b %>% filter(Treatment %in% c("LO", "GM"))
b.ld <- ds.b %>% filter(Treatment %in% c("LO", "DM"))
b.dg <- ds.b %>% filter(Treatment %in% c("DM", "GM"))
x <- wilcox.test(perDffOrig ~ Treatment, data=b.cl, conf.int=TRUE, conf.level=0.95,
                 exact=TRUE)
bl.df <- bind_rows(bl.df, data_frame(Treat1=as.character("Control"),
                                     Treat2=as.character("Lampante"),
                                     Loc.shift=x$estimate,
                                     CI.lwr=x$conf.int[1], CI.upr=x$conf.int[2],
                                     p.value=x$p.value))
x <- wilcox.test(perDffOrig ~ Treatment, data=b.cg, conf.int=TRUE, conf.level=0.95,
                 exact=TRUE)
bl.df <- bind_rows(bl.df, data_frame(Treat1=as.character("Control"),
                                     Treat2=as.character("GM"),
                                     Loc.shift=x$estimate,
                                     CI.lwr=x$conf.int[1], CI.upr=x$conf.int[2],
                                     p.value=x$p.value))
x <- wilcox.test(perDffOrig ~ Treatment, data=b.cd, conf.int=TRUE, conf.level=0.95,
                 exact=TRUE)
bl.df <- bind_rows(bl.df, data_frame(Treat1=as.character("Control"),
                                     Treat2=as.character("DM"),
                                     Loc.shift=x$estimate,
                                     CI.lwr=x$conf.int[1], CI.upr=x$conf.int[2],
                                     p.value=x$p.value))
x <- wilcox.test(perDffOrig ~ Treatment, data=b.lg, conf.int=TRUE, conf.level=0.95,
                 exact=TRUE)
bl.df <- bind_rows(bl.df, data_frame(Treat1=as.character("GM"),
                                     Treat2=as.character("Lampante"),
                                     Loc.shift=x$estimate,
                                     CI.lwr=x$conf.int[1], CI.upr=x$conf.int[2],
                                     p.value=x$p.value))
x <- wilcox.test(perDffOrig ~ Treatment, data=b.ld, conf.int=TRUE, conf.level=0.95,
                 exact=TRUE)
```

```
bl.df <- bind_rows(bl.df, data_frame(Treat1=as.character("DM"),
                                   Treat2=as.character("Lampante"),
                                   Loc.shift=x$estimate,
                                   CI.lwr=x$conf.int[1], CI.upr=x$conf.int[2],
                                   p.value=x$p.value))
x <- wilcox.test(perDffOrig ~ Treatment, data=b.dg, conf.int=TRUE, conf.level=0.95,
                exact=TRUE)
bl.df <- bind_rows(bl.df, data_frame(Treat1=as.character("DM"),
                                   Treat2=as.character("GM"),
                                   Loc.shift=x$estimate,
                                   CI.lwr=x$conf.int[1], CI.upr=x$conf.int[2],
                                   p.value=x$p.value))
bl.df <- bl.df[-1,]
```

Examine the results and plot. We make some new columns for absolute values to simplify reporting and plotting.

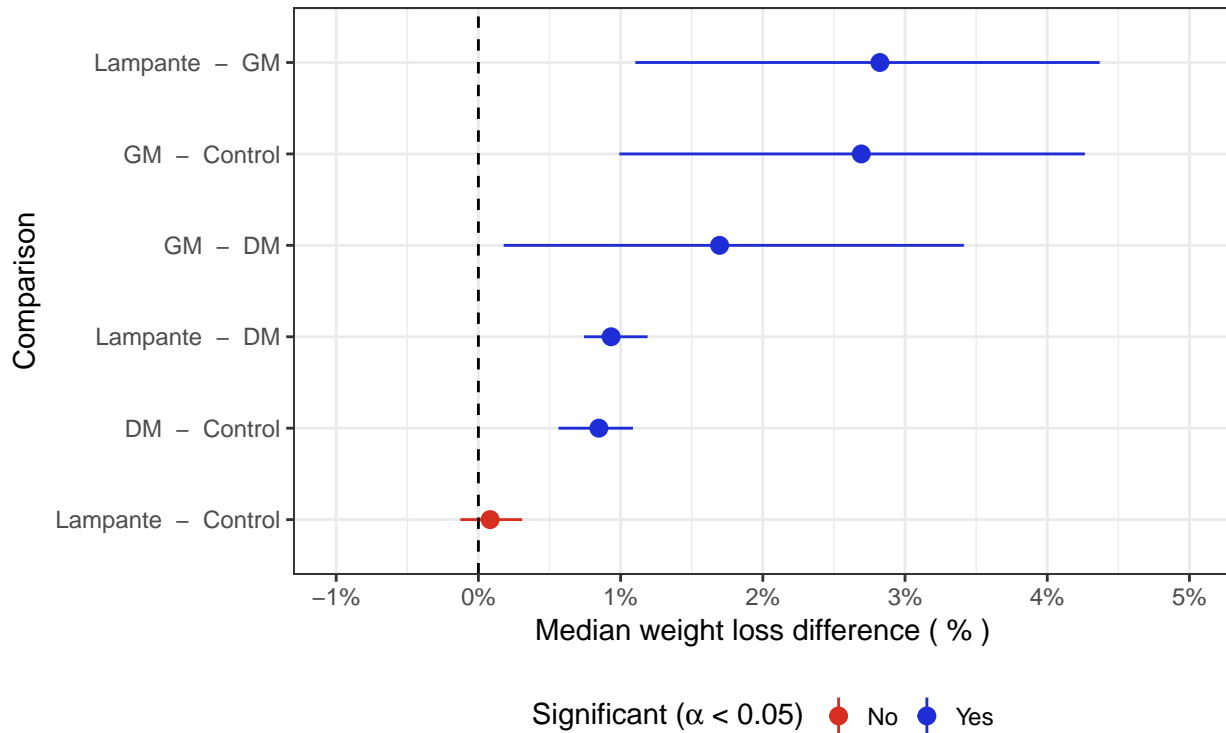
```
bl.df$AbsValue <- abs(bl.df$Loc.shift)
bl.df$AbsCI.upr <- ifelse(bl.df$Loc.shift < 0, bl.df$CI.lwr * -1, bl.df$CI.upr)
bl.df$AbsCI.lwr <- ifelse(bl.df$Loc.shift < 0, bl.df$CI.upr * -1, bl.df$CI.lwr)
# create a nice label for the chart
bl.df$Label <- ifelse(bl.df$Loc.shift > 1, paste(bl.df$Treat1, " - ", bl.df$Treat2),
                    paste(bl.df$Treat2, " - ", bl.df$Treat1))
bl.df$Significant <- ifelse(bl.df$p.value < 0.05, "Yes", "No") #identify sig. differences
bl.df <- bl.df %>% arrange(AbsValue) # order by the magnitude of the effect size (step 1)
levels <- bl.df$Label # step 2
bl.df$Label <- factor(bl.df$Label, levels=levels) #step 3
bl.df %>% select(Label, AbsValue, AbsCI.lwr, AbsCI.upr, Significant, p.value) #inspect
```

Label	AbsValue	AbsCI.lwr	AbsCI.upr	Significant	p.value
Lampante - Control	0.0008228	-0.0012681	0.0030725	No	0.5787417
DM - Control	0.0084726	0.0056336	0.0108687	Yes	0.0000108
Lampante - DM	0.0093325	0.0074273	0.0119005	Yes	0.0000108
GM - DM	0.0169594	0.0017784	0.0341360	Yes	0.0051960
GM - Control	0.0269211	0.0099111	0.0426461	Yes	0.0000108
Lampante - GM	0.0282308	0.0110306	0.0436861	Yes	0.0000108

```
#plot
ggplot(data=bl.df, aes(x=Label, y=AbsValue, ymin=AbsCI.lwr, ymax=AbsCI.upr,
                      colour=Significant)) + theme_bw() +
  geom_pointrange() +
  geom_hline(yintercept=0, linetype=2) +
  scale_colour_manual(values=c("#D62F1F", "#1E2DD6"),
                     name=expression(paste("Significant (", alpha, " < 0.05)")) +
  scale_y_continuous(labels=percent, limits=(c(-0.01, 0.05)),
                    breaks=seq(-0.01, 0.05, 0.01)) +
  coord_flip() +
  labs(title="Leaching: Treatment differences for Beech",
       subtitle="Median difference bewtween samples, with 95 % CIs",
       x="Comparison", y="Median weight loss difference ( % )") +
  theme(legend.position = "bottom")
```

## Leaching: Treatment differences for Beech

Median difference between samples, with 95 % CIs



## Pine treatment comparisons

\*\* need to check abs ci vs normal ci and compare to beech \*\*

```
pl.df <- data_frame(Treat1=as.character("treat1"), Treat2=as.character("treat2"),
  Loc.shift = 0, CI.lwr = 0, CI.upr = 0, p.value = 0)
p.cl <- ds.p %>% filter(Treatment %in% c("Control", "LO"))
p.cg <- ds.p %>% filter(Treatment %in% c("Control", "GM"))
p.cd <- ds.p %>% filter(Treatment %in% c("Control", "DM"))
p.lg <- ds.p %>% filter(Treatment %in% c("LO", "GM"))
p.ld <- ds.p %>% filter(Treatment %in% c("LO", "DM"))
p.dg <- ds.p %>% filter(Treatment %in% c("DM", "GM"))
x <- wilcox.test(perDffOrig ~ Treatment, data=p.cl, conf.int=TRUE, conf.level=0.95,
  exact=TRUE)
pl.df <- bind_rows(pl.df, data_frame(Treat1=as.character("Control"),
  Treat2=as.character("Lampante"),
  Loc.shift=x$estimate,
  CI.lwr=x$conf.int[1], CI.upr=x$conf.int[2],
  p.value=x$p.value))
x <- wilcox.test(perDffOrig ~ Treatment, data=p.cg, conf.int=TRUE, conf.level=0.95,
  exact=TRUE)
pl.df <- bind_rows(pl.df, data_frame(Treat1=as.character("Control"),
  Treat2=as.character("GM"),
  Loc.shift=x$estimate,
  CI.lwr=x$conf.int[1], CI.upr=x$conf.int[2],
  p.value=x$p.value))
x <- wilcox.test(perDffOrig ~ Treatment, data=p.cd, conf.int=TRUE, conf.level=0.95,
```

```

        exact=TRUE)
pl.df <- bind_rows(pl.df, data_frame(Treat1=as.character("Control"),
                                   Treat2=as.character("DM"),
                                   Loc.shift=x$estimate,
                                   CI.lwr=x$conf.int[1], CI.upr=x$conf.int[2],
                                   p.value=x$p.value))
x <- wilcox.test(perDffOrig ~ Treatment, data=p.lg, conf.int=TRUE, conf.level=0.95,
                exact=TRUE)
pl.df <- bind_rows(pl.df, data_frame(Treat1=as.character("GM"),
                                   Treat2=as.character("Lampante"),
                                   Loc.shift=x$estimate,
                                   CI.lwr=x$conf.int[1], CI.upr=x$conf.int[2],
                                   p.value=x$p.value))
x <- wilcox.test(perDffOrig ~ Treatment, data=p.ld, conf.int=TRUE, conf.level=0.95,
                exact=TRUE)
pl.df <- bind_rows(pl.df, data_frame(Treat1=as.character("DM"),
                                   Treat2=as.character("Lampante"),
                                   Loc.shift=x$estimate,
                                   CI.lwr=x$conf.int[1], CI.upr=x$conf.int[2],
                                   p.value=x$p.value))
x <- wilcox.test(perDffOrig ~ Treatment, data=p.dg, conf.int=TRUE, conf.level=0.95,
                exact=TRUE)
pl.df <- bind_rows(pl.df, data_frame(Treat1=as.character("DM"),
                                   Treat2=as.character("GM"),
                                   Loc.shift=x$estimate,
                                   CI.lwr=x$conf.int[1], CI.upr=x$conf.int[2],
                                   p.value=x$p.value))

pl.df <- pl.df[-1,]

```

Examine the Results and plot.

```

pl.df$AbsValue <- abs(pl.df$Loc.shift)
pl.df$AbsCI.upr <- ifelse(pl.df$Loc.shift < 0, pl.df$CI.lwr * -1, pl.df$CI.upr)
pl.df$AbsCI.lwr <- ifelse(pl.df$Loc.shift < 0, pl.df$CI.upr * -1, pl.df$CI.lwr)
# create a nice label for the chart
pl.df$Label <- ifelse(pl.df$Loc.shift > 1, paste(pl.df$Treat1, " - ", pl.df$Treat2),
                    paste(pl.df$Treat2, " - ", pl.df$Treat1))
pl.df$Significant <- ifelse(pl.df$p.value < 0.05, "Yes", "No") #identify sig. differences
pl.df <- pl.df %>% arrange(AbsValue) # order by the magnitude of the effect size (step 1)
levels <- pl.df$Label # step 2
pl.df$Label <- factor(pl.df$Label, levels=levels) #step 3
pl.df %>% select(Label, AbsValue, AbsCI.lwr, AbsCI.upr, Significant, p.value) #inspect

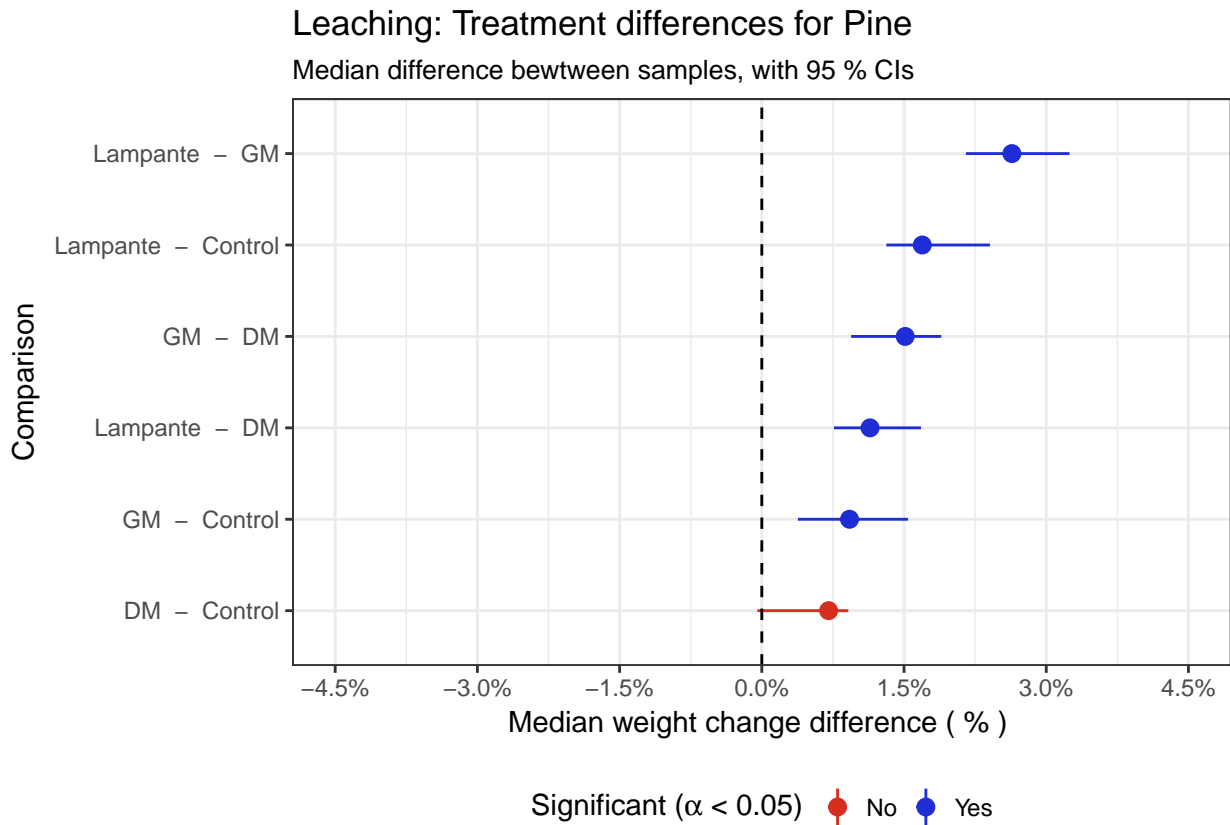
```

Label	AbsValue	AbsCI.lwr	AbsCI.upr	Significant	p.value
DM - Control	0.0070472	-0.0004737	0.0091208	No	0.0630128
GM - Control	0.0092512	0.0038057	0.0154269	Yes	0.0068415
Lampante - DM	0.0114136	0.0076101	0.0167853	Yes	0.0000108
GM - DM	0.0151215	0.0094223	0.0189241	Yes	0.0000108
Lampante - Control	0.0169232	0.0131353	0.0240688	Yes	0.0000217
Lampante - GM	0.0263761	0.0215237	0.0324588	Yes	0.0000108

```

ggplot(data=pl.df, aes(x=Label, y=AbsValue, ymin=AbsCI.lwr, ymax=AbsCI.upr,
                      colour=Significant)) + theme_bw() +
  geom_pointrange() +
  geom_hline(yintercept=0, linetype=2) +
  scale_colour_manual(values=c("#D62F1F", "#1E2DD6"),
                    name=expression(paste("Significant (", alpha, " < 0.05)"))) +
  scale_y_continuous(labels=percent, limits=c(-0.045, 0.045),
                    breaks=seq(-0.045, 0.045, 0.015)) +
  coord_flip() +
  labs(title="Leaching: Treatment differences for Pine",
       subtitle="Median difference bewtween samples, with 95 % CIs",
       x="Comparison", y="Median weight change difference ( % ) ") +
  theme(legend.position = "bottom")

```



# Fungi Results

## Read and prepare the data

```
fg <- read.csv("data/OOTFungi1MS_11112017_v1_r0.csv", stringsAsFactors=FALSE)
fg <- fg %>% separate(SpecimenID, c("Species", "Treatment", "Test", "SpecNo"),
  sep="-", remove=FALSE, fill="right", extra="warn")

fg$Test <- "MB" #mini block

fg$SpecNo <- ifelse(fg$Treatment == "kontrola" & fg$Species == "P", seq(1,40,1),
  ifelse(fg$Treatment == "kontrola" & fg$Species == "B", seq(1,40,1),
  fg$Number))

fg$EWWW_per <- fg$EndWeightMinusWoodWeight / fg$WoodWeight

fg$Treatment <- ifelse(fg$Treatment == "kontrola", "Control", fg$Treatment)

fg$MassLoss <- (fg$InitialDryWeight - fg$PostInnocDryWeight) #Mass loss in grams
fg$InocLongName <- ifelse(fg$Inoculant == "gt", "Gloeopyllum trabeum",
  ifelse(fg$Inoculant == "pl", "Pleurotus ostreatus",
  ifelse(fg$Inoculant == "pm", "Poria monticola",
  ifelse(fg$Inoculant == "tv", "Trametes versicolor", "Zed"))))

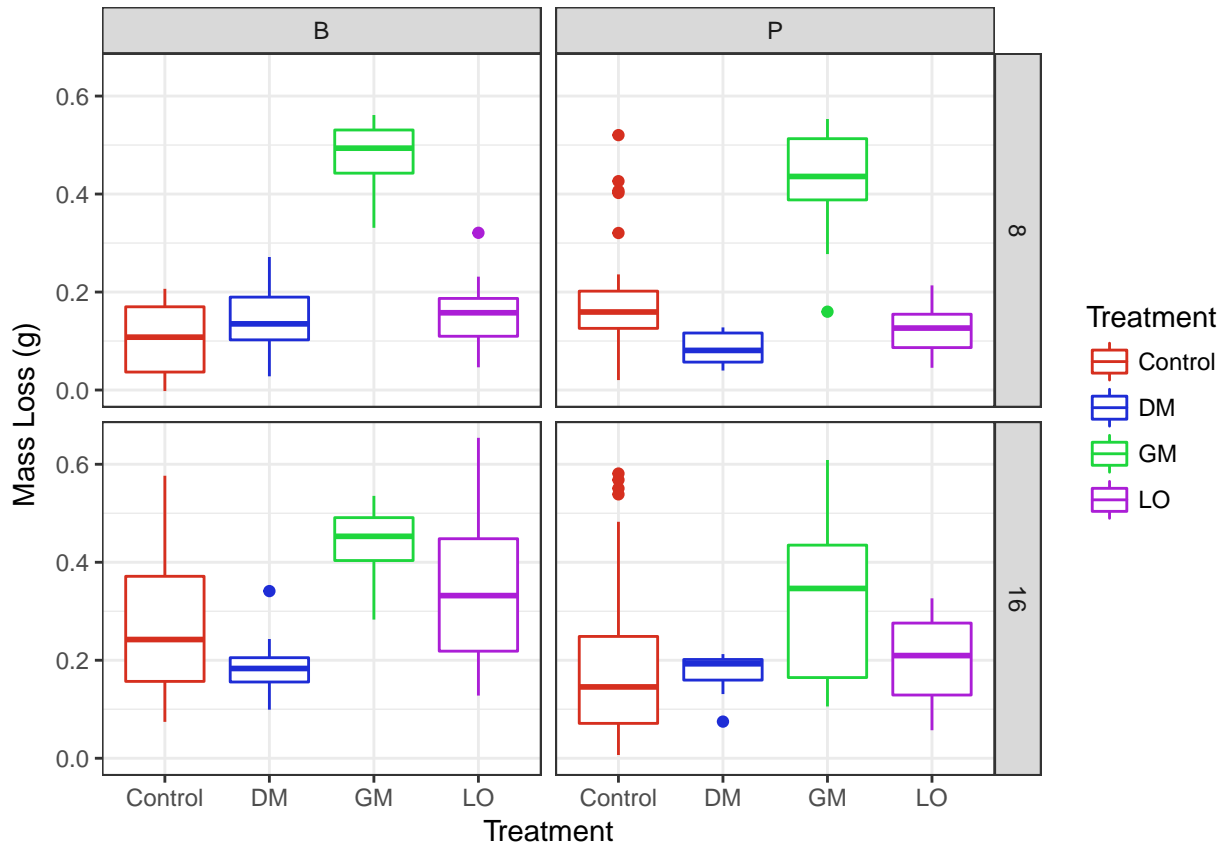
glimpse(fg)
```

```
## Observations: 384
## Variables: 29
## $ SpecimenID      <chr> "B-DM-MB-1", "B-DM-MB-10", "B-DM-MB-1...
## $ Species         <chr> "B", "B", "B", "B", "B", "B", "B", "B...
## $ Treatment       <chr> "DM", "DM", "DM", "DM", "DM", "DM", "...
## $ Test            <chr> "MB", "MB", "MB", "MB", "MB", "MB", "...
## $ SpecNo          <dbl> 1, 10, 11, 12, 13, 14, 15, 16, 17, 18...
## $ Number          <int> 1, 10, 11, 12, 13, 14, 15, 16, 17, 18...
## $ StartDate       <chr> "13/07/2016", "13/07/2016", "13/07/20...
## $ EndDate         <chr> "02/11/2016", "02/11/2016", "02/11/20...
## $ Duration        <int> 16, 16, 16, 16, 16, 16, 16, 16, 1...
## $ Inoculant       <chr> "pl", "tv", "tv", "tv", "tv", "tv", "...
## $ InitialDryWeight <dbl> 1.6776, 1.6966, 1.6703, 1.6715, 1.694...
## $ PostInoculationWeight <dbl> 2.1112, 2.1996, 2.1793, 2.1049, 2.120...
## $ PostInnocDryWeight <dbl> 1.5016, 1.4532, 1.4813, 1.5152, 1.480...
## $ PostExtractionWeight <dbl> 1.1203, 1.1034, 1.0812, 1.1111, 1.077...
## $ ImpregnatedOilWeight <dbl> 0.648, 0.681, 0.697, 0.654, 0.673, 0...
## $ Retention.kg.m3 <int> 432, 454, 465, 436, 449, 438, 445, 44...
## $ WoodWeight      <dbl> 1.0296, 1.0156, 0.9733, 1.0175, 1.021...
## $ RemainingOilWeight.g <dbl> 0.4720, 0.4376, 0.5080, 0.4977, 0.459...
## $ EndWeightMinusWoodWeight <dbl> 0.0907, 0.0878, 0.1079, 0.0936, 0.055...
## $ OilRemainingPerc <dbl> 0.73, 0.64, 0.73, 0.76, 0.68, 0.72, 0...
## $ MCPostInoculation <dbl> 0.41, 0.51, 0.47, 0.39, 0.43, 0.43, 0...
## $ RelativeMassLoss <dbl> 0.10, 0.14, 0.11, 0.09, 0.13, 0.11, 0...
## $ OilExtract.g    <dbl> 0.38, 0.35, 0.40, 0.40, 0.40, 0.39, 0...
## $ OilExtract_rel  <dbl> 0.25, 0.24, 0.27, 0.27, 0.27, 0.26, 0...
## $ OilContent      <dbl> 0.34, 0.32, 0.37, 0.36, 0.37, 0.36, 0...
## $ PetriDish       <chr> "plo1", "plo2", "plo3", "plo4", "plo5...
## $ EWWW_per        <dbl> 0.08809246, 0.08645136, 0.11085996, 0...
```

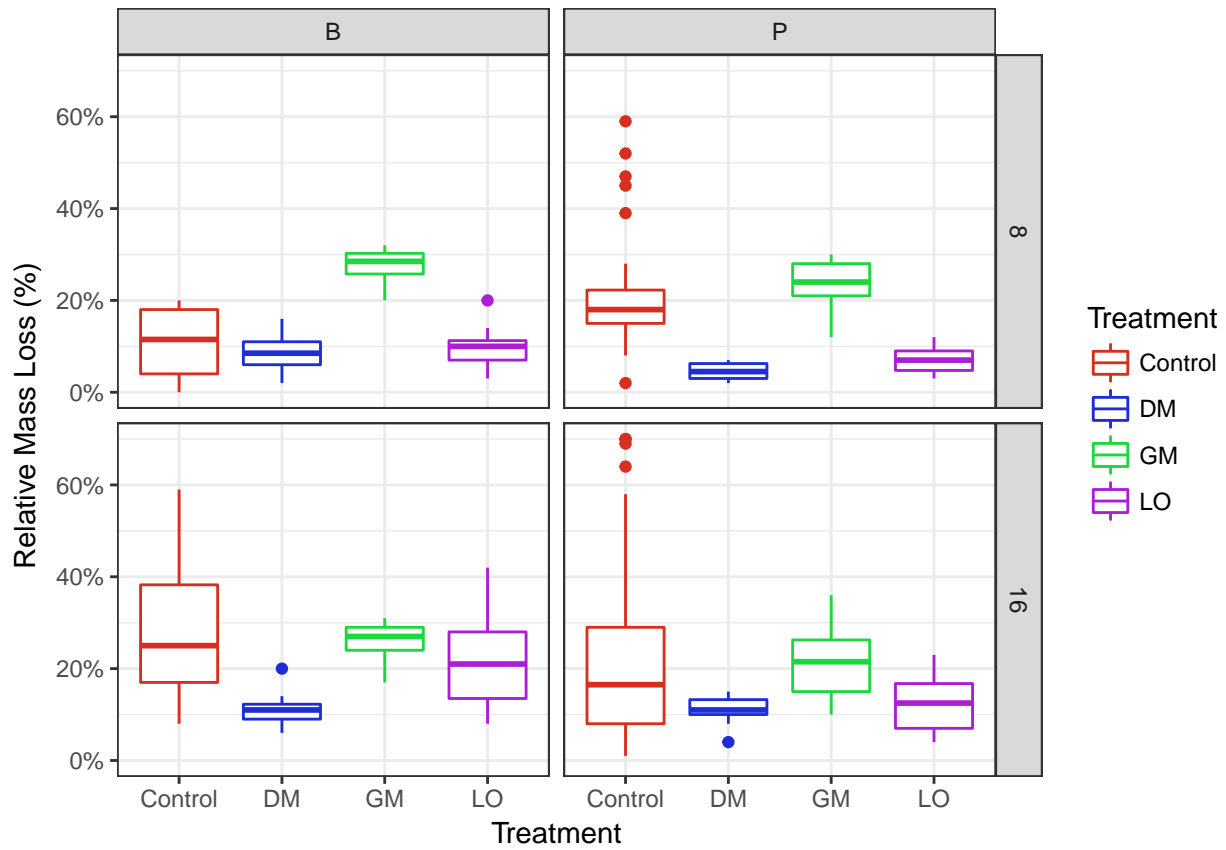


```
## $ MassLoss          <dbl> 0.1760, 0.2434, 0.1890, 0.1563, 0.214...
## $ InocLongName     <chr> "Pleurotus ostreatus", "Trametes vers..."
```

```
ggplot(data=fg, aes(y=MassLoss, x=Treatment, colour=Treatment)) + theme_bw() +
  geom_boxplot() +
  scale_colour_manual(values=c("#D62F1F", "#1E2DD6", "#1ED63D", "#AB1ED6")) +
  facet_grid(Duration~Species) +
  labs(y="Mass Loss (g)")
```



```
ggplot(data=fg, aes(y=RelativeMassLoss, x=Treatment, colour=Treatment)) + theme_bw() +
  geom_boxplot() +
  scale_colour_manual(values=c("#D62F1F", "#1E2DD6", "#1ED63D", "#AB1ED6")) +
  scale_y_continuous(labels=percent) +
  facet_grid(Duration~Species) +
  labs(y="Relative Mass Loss (%)")
```

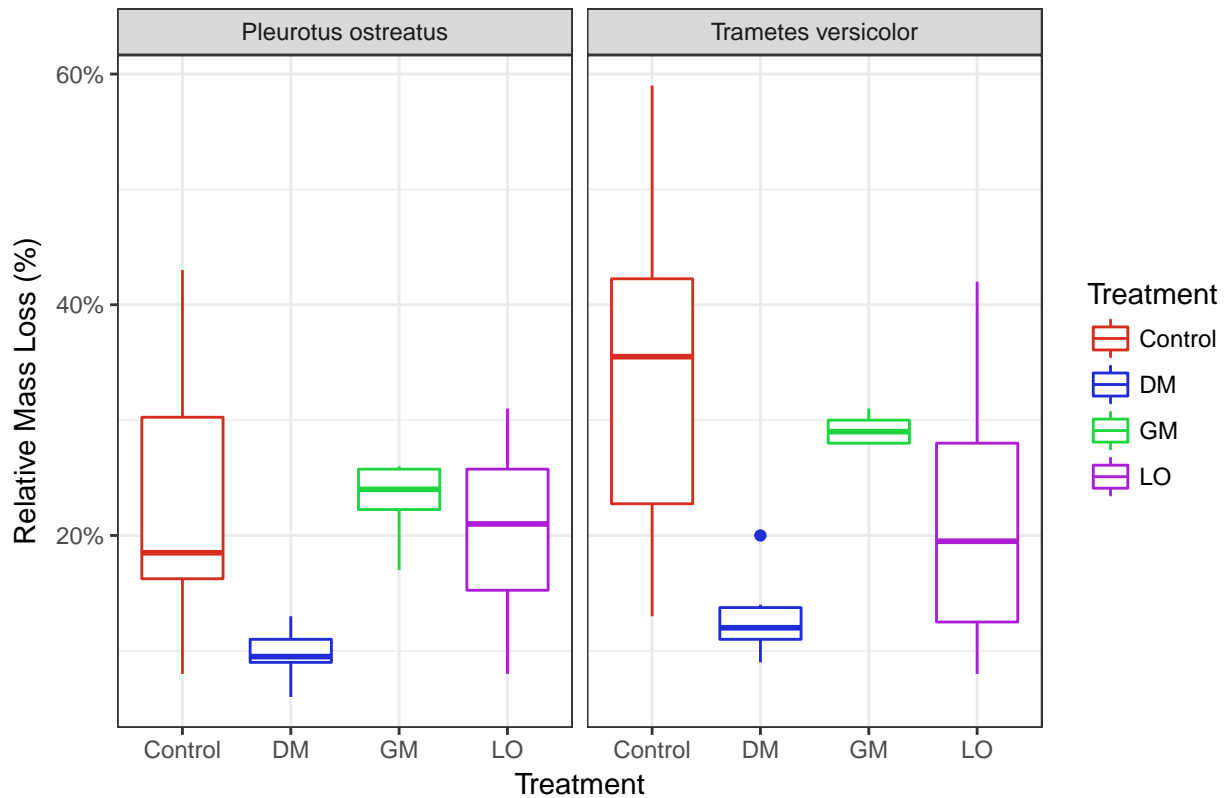


```

fg.b16 <- fg %>% filter(Species == "B", Duration == 16)
ggplot(data=fg.b16, aes(y=RelativeMassLoss, x=Treatment, colour=Treatment)) + theme_bw() +
  geom_boxplot() +
  facet_grid(.~InocLongName) +
  scale_colour_manual(values=c("#D62F1F", "#1E2DD6", "#1ED63D", "#AB1ED6")) +
  scale_y_continuous(labels=percent) +
  labs(title="Beech by inoculant (16 weeks)") +
  labs(y="Relative Mass Loss (%)")

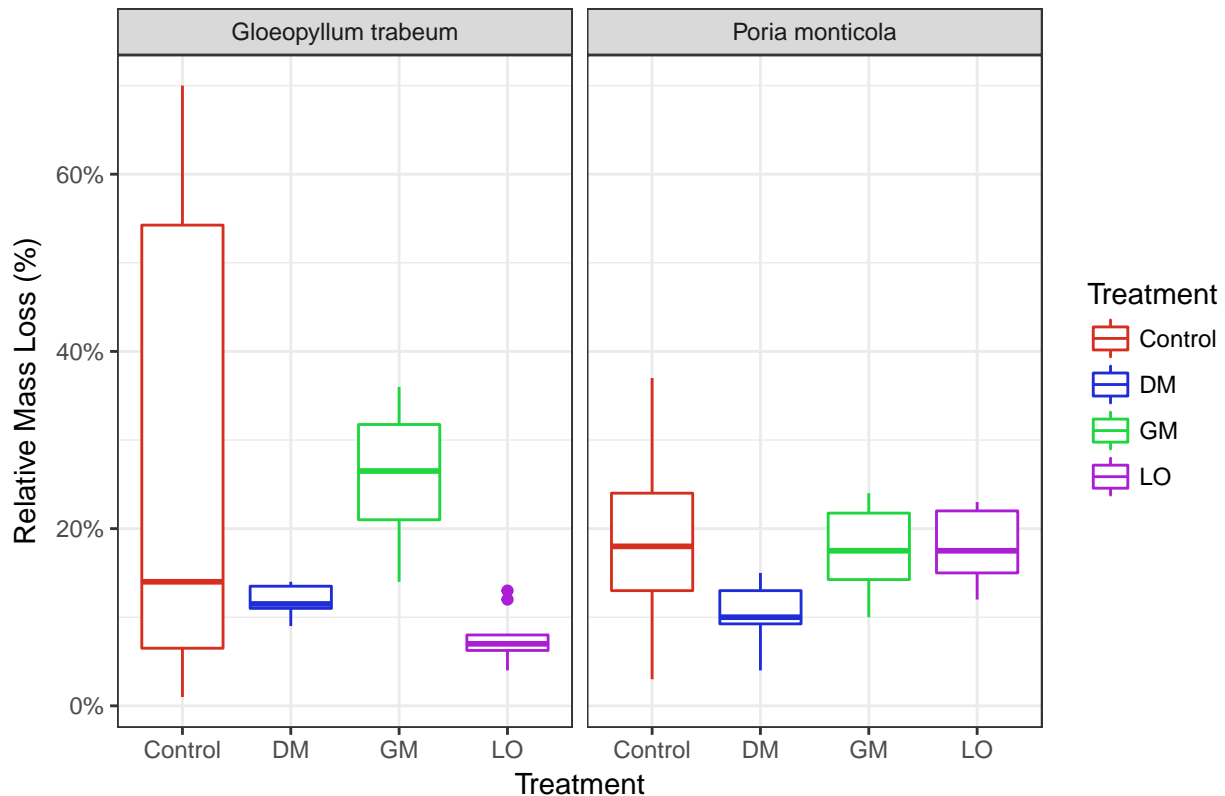
```

## Beech by inoculant (16 weeks)



```
fg.p16 <- fg %>% filter(Species == "P", Duration == 16)
ggplot(data=fg.p16, aes(y=RelativeMassLoss, x=Treatment, colour=Treatment)) + theme_bw() +
  geom_boxplot() +
  facet_grid(.~InocLongName) +
  scale_colour_manual(values=c("#D62F1F", "#1E2DD6", "#1ED63D", "#AB1ED6")) +
  scale_y_continuous(labels=percent) +
  labs(title="Pine by inoculant (16 weeks)") +
  labs(y="Relative Mass Loss (%)")
```

## Pine by inoculant (16 weeks)



## Data summary

Our indicator of choice is Relative Mass Loss (InitialDryWeight - PostInoculantDryWeight) / InitialDryWeight,

```
fg %>% group_by(Species, Treatment) %>%
  summarise(RelativeMassLoss.mu = mean(RelativeMassLoss),
            RelativeMassLoss.med = median(RelativeMassLoss),
            RelativeMassLoss.sd = sd(RelativeMassLoss), n = n())
```

Species	Treatment	RelativeMassLoss.mu	RelativeMassLoss.med	RelativeMassLoss.sd	n
B	Control	0.1958333	0.175	0.1407800	72
B	DM	0.0982500	0.100	0.0335802	40
B	GM	0.2687500	0.280	0.0390389	40
B	LO	0.1532500	0.120	0.0933257	40
P	Control	0.2188889	0.180	0.1678325	72
P	DM	0.0800000	0.070	0.0392232	40
P	GM	0.2280000	0.225	0.0646965	40
P	LO	0.0977500	0.080	0.0574004	40

## Comparisons

Here we compare the difference in *RelativeMassLoss* between treatments for each species and Inoculant.

## Beech :: Inoculant PL

First, we compare responses to the PL Inoculant for beech at 16 weeks.

```
cat("Beech :: PL")

## Beech :: PL

bpl <- fg.b16 %>% filter(Inoculant == "pl")
kruskal.test(bpl$RelativeMassLoss, as.factor(bpl$Treatment))

##
## Kruskal-Wallis rank sum test
##
## data: bpl$RelativeMassLoss and as.factor(bpl$Treatment)
## Kruskal-Wallis chi-squared = 17.974, df = 3, p-value = 0.0004452
```

The p-value indicates there is some difference between treatments, so we check them all with wilcoxon rank sum tests.

```
bpl.df <- data_frame(Treat1=as.character("treat1"), Treat2=as.character("treat2"),
                    Loc.shift = 0, CI.lwr = 0, CI.upr = 0, p.value = 0)
bpl.c1 <- bpl %>% filter(Treatment %in% c("Control", "LO"))
bpl.cd <- bpl %>% filter(Treatment %in% c("Control", "DM"))
bpl.cg <- bpl %>% filter(Treatment %in% c("Control", "GM"))
bpl.ld <- bpl %>% filter(Treatment %in% c("LO", "DM"))
bpl.lg <- bpl %>% filter(Treatment %in% c("LO", "GM"))
bpl.dg <- bpl %>% filter(Treatment %in% c("DM", "GM"))

x <- wilcox.test(RelativeMassLoss~as.factor(Treatment), data=bpl.c1, conf.int=TRUE,
                conf.level=0.95, exact=TRUE)

## Warning in wilcox.test.default(x = c(0.24, 0.15, 0.08, 0.19, 0.31, 0.17, :
## cannot compute exact p-value with ties

## Warning in wilcox.test.default(x = c(0.24, 0.15, 0.08, 0.19, 0.31, 0.17, :
## cannot compute exact confidence intervals with ties

bpl.df <- bind_rows(bpl.df, data_frame(Treat1=as.character("Control"),
                                       Treat2=as.character("Lampante"),
                                       Loc.shift=x$estimate,
                                       CI.lwr=x$conf.int[1], CI.upr=x$conf.int[2],
                                       p.value=x$p.value))
x <- wilcox.test(RelativeMassLoss~as.factor(Treatment), data=bpl.cd, conf.int=TRUE,
                conf.level=0.95, exact=TRUE)
```

```
## Warning in wilcox.test.default(x = c(0.24, 0.15, 0.08, 0.19, 0.31, 0.17, :
## cannot compute exact p-value with ties

## Warning in wilcox.test.default(x = c(0.24, 0.15, 0.08, 0.19, 0.31, 0.17, :
## cannot compute exact confidence intervals with ties

bpl.df <- bind_rows(bpl.df, data_frame(Treat1=as.character("Control"),
                                       Treat2=as.character("DM"),
                                       Loc.shift=x$estimate,
                                       CI.lwr=x$conf.int[1], CI.upr=x$conf.int[2],
                                       p.value=x$p.value))
x <- wilcox.test(RelativeMassLoss~as.factor(Treatment), data=bpl.cg, conf.int=TRUE,
                conf.level=0.95, exact=TRUE)
```

```

## Warning in wilcox.test.default(x = c(0.24, 0.15, 0.08, 0.19, 0.31, 0.17, :
## cannot compute exact p-value with ties

## Warning in wilcox.test.default(x = c(0.24, 0.15, 0.08, 0.19, 0.31, 0.17, :
## cannot compute exact confidence intervals with ties

bpl.df <- bind_rows(bpl.df, data_frame(Treat1=as.character("Control"),
                                     Treat2=as.character("GM"),
                                     Loc.shift=x$estimate,
                                     CI.lwr=x$conf.int[1], CI.upr=x$conf.int[2],
                                     p.value=x$p.value))
x <- wilcox.test(RelativeMassLoss~as.factor(Treatment), data=bpl.ld, conf.int=TRUE,
                conf.level=0.95, exact=TRUE)

```

```

## Warning in wilcox.test.default(x = c(0.1, 0.06, 0.13, 0.11, 0.09, 0.12, :
## cannot compute exact p-value with ties

## Warning in wilcox.test.default(x = c(0.1, 0.06, 0.13, 0.11, 0.09, 0.12, :
## cannot compute exact confidence intervals with ties

bpl.df <- bind_rows(bpl.df, data_frame(Treat1=as.character("DM"),
                                     Treat2=as.character("Lampante"),
                                     Loc.shift=x$estimate,
                                     CI.lwr=x$conf.int[1], CI.upr=x$conf.int[2],
                                     p.value=x$p.value))
x <- wilcox.test(RelativeMassLoss~as.factor(Treatment), data=bpl.lg, conf.int=TRUE,
                conf.level=0.95, exact=TRUE)

```

```

## Warning in wilcox.test.default(x = c(0.23, 0.24, 0.26, 0.17, 0.26, 0.18, :
## cannot compute exact p-value with ties

## Warning in wilcox.test.default(x = c(0.23, 0.24, 0.26, 0.17, 0.26, 0.18, :
## cannot compute exact confidence intervals with ties

bpl.df <- bind_rows(bpl.df, data_frame(Treat1=as.character("GM"),
                                     Treat2=as.character("Lampante"),
                                     Loc.shift=x$estimate,
                                     CI.lwr=x$conf.int[1], CI.upr=x$conf.int[2],
                                     p.value=x$p.value))
x <- wilcox.test(RelativeMassLoss~as.factor(Treatment), data=bpl.dg, conf.int=TRUE,
                conf.level=0.95, exact=TRUE)

```

```

## Warning in wilcox.test.default(x = c(0.1, 0.06, 0.13, 0.11, 0.09, 0.12, :
## cannot compute exact p-value with ties

## Warning in wilcox.test.default(x = c(0.1, 0.06, 0.13, 0.11, 0.09, 0.12, :
## cannot compute exact confidence intervals with ties

bpl.df <- bind_rows(bpl.df, data_frame(Treat1=as.character("DM"),
                                     Treat2=as.character("GM"),
                                     Loc.shift=x$estimate,
                                     CI.lwr=x$conf.int[1], CI.upr=x$conf.int[2],
                                     p.value=x$p.value))

bpl.df <- bpl.df[-1,]

```

Examine the results and plot.

```

bpl.df$AbsValue <- abs(bpl.df$Loc.shift)
bpl.df$AbsCI.upr <- ifelse(bpl.df$Loc.shift < 0, bpl.df$CI.lwr * -1, bpl.df$CI.upr)

```

```

bpl.df$AbsCI.lwr <- ifelse(bpl.df$Loc.shift < 0, bpl.df$CI.upr * -1, bpl.df$CI.lwr)
# create a nice label for the chart
bpl.df$Label <- ifelse(bpl.df$Loc.shift > 1, paste(bpl.df$Treat1, " - ", bpl.df$Treat2),
  paste(bpl.df$Treat2, " - ", bpl.df$Treat1))
bpl.df$Significant <- ifelse(bpl.df$p.value < 0.05, "Yes", "No") #identify sig. differences
bpl.df <- bpl.df %>% arrange(AbsValue) # order by the magnitude of the effect size (step 1)
levels <- bpl.df$Label # step 2
bpl.df$Label <- factor(bpl.df$Label, levels=levels) #step 3
bpl.df %>% select(Label, AbsValue, AbsCI.lwr, AbsCI.upr, Significant, p.value) #inspect

```

Label	AbsValue	AbsCI.lwr	AbsCI.upr	Significant	p.value
Lampante - Control	0.0132501	-0.0600119	0.0999591	No	0.8288992
GM - Control	0.0200209	-0.0699660	0.0799849	No	0.4136128
Lampante - GM	0.0241753	-0.0399570	0.0900499	No	0.5184717
DM - Control	0.0999756	0.0600335	0.1999820	Yes	0.0004165
Lampante - DM	0.1199821	0.0400426	0.1700209	Yes	0.0034978
GM - DM	0.1399665	0.1100440	0.1600225	Yes	0.0001727

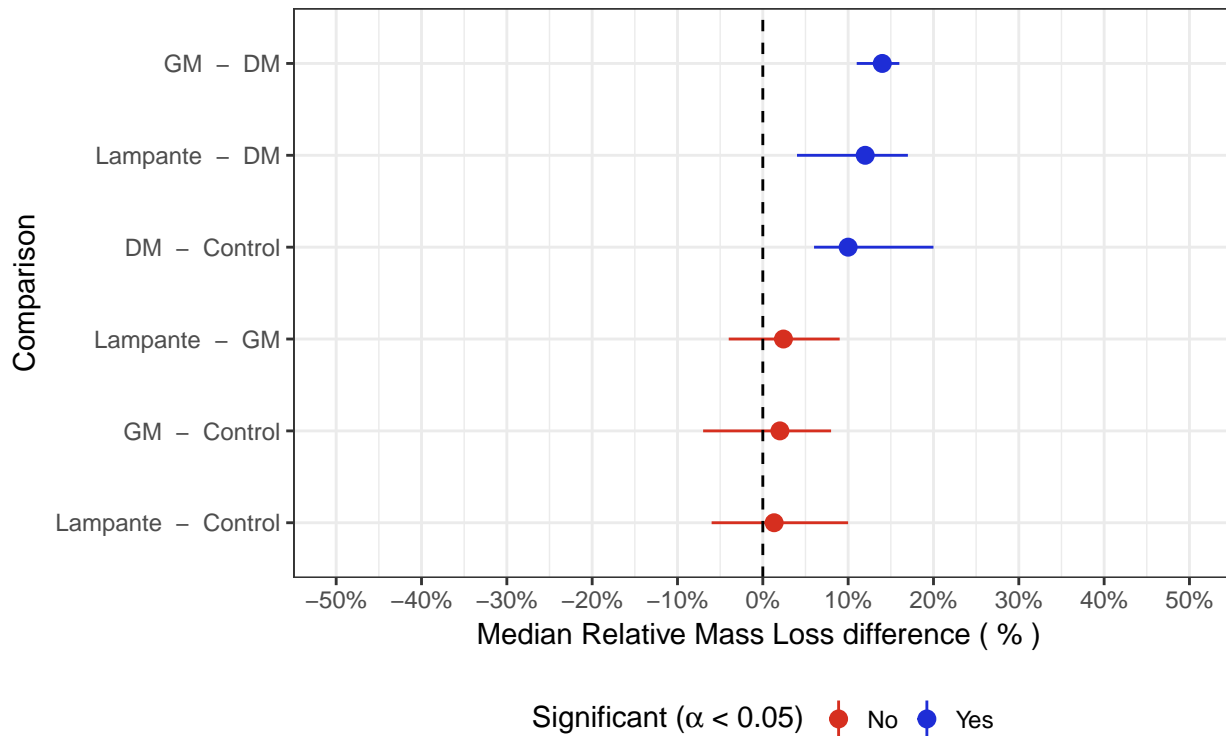
```

ggplot(data=bpl.df, aes(x=Label, y=AbsValue, ymin=AbsCI.lwr, ymax=AbsCI.upr,
  colour=Significant)) + theme_bw() +
  geom_pointrange() +
  geom_hline(yintercept=0, linetype=2) +
  scale_colour_manual(values=c("#D62F1F", "#1E2DD6"),
    name=expression(paste("Significant (", alpha, " < 0.05)")) +
  scale_y_continuous(labels=percent, limits=(c(-0.5, 0.5)), breaks=seq(-0.5, 0.5, 0.1)) +
  coord_flip() +
  labs(title="PL Inoculant, Beech",
    subtitle="Median difference bewtween samples, with 95 % CIs",
    x="Comparison", y="Median Relative Mass Loss difference ( % ) ") +
  theme(legend.position = "bottom")

```

## PL Inoculant, Beech

Median difference between samples, with 95 % CIs



## Beech :: Inoculant TV

```
btv <- fg.b16 %>% filter(Inoculant == "tv")
kruskal.test(btv$RelativeMassLoss, as.factor(btv$Treatment))
```

```
##
## Kruskal-Wallis rank sum test
##
## data: btv$RelativeMassLoss and as.factor(btv$Treatment)
## Kruskal-Wallis chi-squared = 21.516, df = 3, p-value = 8.226e-05
```

The p-value here suggests a difference between at least two of the treatments, so we check them all.

```
btv.df <- data_frame(Treat1=as.character("treat1"), Treat2=as.character("treat2"),
  Loc.shift = 0, CI.lwr = 0, CI.upr = 0, p.value = 0)
btv.c1 <- btv %>% filter(Treatment %in% c("Control", "LO"))
btv.cd <- btv %>% filter(Treatment %in% c("Control", "DM"))
btv.cg <- btv %>% filter(Treatment %in% c("Control", "GM"))
btv.ld <- btv %>% filter(Treatment %in% c("LO", "DM"))
btv.lg <- btv %>% filter(Treatment %in% c("LO", "GM"))
btv.dg <- btv %>% filter(Treatment %in% c("DM", "GM"))
x <- wilcox.test(RelativeMassLoss~as.factor(Treatment), data=btv.c1, conf.int=TRUE,
  conf.level=0.95, exact=TRUE)
```

```
## Warning in wilcox.test.default(x = c(0.27, 0.14, 0.25, 0.15, 0.58, 0.59, :
## cannot compute exact p-value with ties
```

```
## Warning in wilcox.test.default(x = c(0.27, 0.14, 0.25, 0.15, 0.58, 0.59, :
```



```

## cannot compute exact confidence intervals with ties
btv.df <- bind_rows(btv.df, data_frame(Treat1=as.character("Control"),
                                     Treat2=as.character("Lampante"),
                                     Loc.shift=x$estimate,
                                     CI.lwr=x$conf.int[1], CI.upr=x$conf.int[2],
                                     p.value=x$p.value))
x <- wilcox.test(RelativeMassLoss~as.factor(Treatment), data=btv.cd, conf.int=TRUE,
                conf.level=0.95, exact=TRUE)

```

```

## Warning in wilcox.test.default(x = c(0.27, 0.14, 0.25, 0.15, 0.58, 0.59, :
## cannot compute exact p-value with ties

```

```

## Warning in wilcox.test.default(x = c(0.27, 0.14, 0.25, 0.15, 0.58, 0.59, :
## cannot compute exact confidence intervals with ties

```

```

btv.df <- bind_rows(btv.df, data_frame(Treat1=as.character("Control"),
                                     Treat2=as.character("DM"),
                                     Loc.shift=x$estimate,
                                     CI.lwr=x$conf.int[1], CI.upr=x$conf.int[2],
                                     p.value=x$p.value))
x <- wilcox.test(RelativeMassLoss~as.factor(Treatment), data=btv.cg, conf.int=TRUE,
                conf.level=0.95, exact=TRUE)

```

```

## Warning in wilcox.test.default(x = c(0.27, 0.14, 0.25, 0.15, 0.58, 0.59, :
## cannot compute exact p-value with ties

```

```

## Warning in wilcox.test.default(x = c(0.27, 0.14, 0.25, 0.15, 0.58, 0.59, :
## cannot compute exact confidence intervals with ties

```

```

btv.df <- bind_rows(btv.df, data_frame(Treat1=as.character("Control"),
                                     Treat2=as.character("GM"),
                                     Loc.shift=x$estimate,
                                     CI.lwr=x$conf.int[1], CI.upr=x$conf.int[2],
                                     p.value=x$p.value))
x <- wilcox.test(RelativeMassLoss~as.factor(Treatment), data=btv.ld, conf.int=TRUE,
                conf.level=0.95, exact=TRUE)

```

```

## Warning in wilcox.test.default(x = c(0.14, 0.11, 0.09, 0.13, 0.11, 0.2, :
## cannot compute exact p-value with ties

```

```

## Warning in wilcox.test.default(x = c(0.14, 0.11, 0.09, 0.13, 0.11, 0.2, :
## cannot compute exact confidence intervals with ties

```

```

btv.df <- bind_rows(btv.df, data_frame(Treat1=as.character("DM"),
                                     Treat2=as.character("Lampante"),
                                     Loc.shift=x$estimate,
                                     CI.lwr=x$conf.int[1], CI.upr=x$conf.int[2],
                                     p.value=x$p.value))
x <- wilcox.test(RelativeMassLoss~as.factor(Treatment), data=btv.lg, conf.int=TRUE,
                conf.level=0.95, exact=TRUE)

```

```

## Warning in wilcox.test.default(x = c(0.3, 0.28, 0.28, 0.3, 0.29, 0.31,
## 0.28, : cannot compute exact p-value with ties

```

```

## Warning in wilcox.test.default(x = c(0.3, 0.28, 0.28, 0.3, 0.29, 0.31,
## 0.28, : cannot compute exact confidence intervals with ties

```

```
btv.df <- bind_rows(btv.df, data_frame(Treat1=as.character("GM"),
                                     Treat2=as.character("Lampante"),
                                     Loc.shift=x$estimate,
                                     CI.lwr=x$conf.int[1], CI.upr=x$conf.int[2],
                                     p.value=x$p.value))
x <- wilcox.test(RelativeMassLoss~as.factor(Treatment), data=btv.dg, conf.int=TRUE,
                conf.level=0.95, exact=TRUE)
```

```
## Warning in wilcox.test.default(x = c(0.14, 0.11, 0.09, 0.13, 0.11, 0.2, :
## cannot compute exact p-value with ties
```

```
## Warning in wilcox.test.default(x = c(0.14, 0.11, 0.09, 0.13, 0.11, 0.2, :
## cannot compute exact confidence intervals with ties
```

```
btv.df <- bind_rows(btv.df, data_frame(Treat1=as.character("DM"),
                                     Treat2=as.character("GM"),
                                     Loc.shift=x$estimate,
                                     CI.lwr=x$conf.int[1], CI.upr=x$conf.int[2],
                                     p.value=x$p.value))

btv.df <- btv.df[-1,]
```

Examine the results and plot.

```
btv.df$AbsValue <- abs(btv.df$Loc.shift)
btv.df$AbsCI.upr <- ifelse(btv.df$Loc.shift < 0, btv.df$CI.lwr * -1, btv.df$CI.upr)
btv.df$AbsCI.lwr <- ifelse(btv.df$Loc.shift < 0, btv.df$CI.upr * -1, btv.df$CI.lwr)
# create a nice label for the chart
btv.df$Label <- ifelse(btv.df$Loc.shift > 1, paste(btv.df$Treat1, " - ", btv.df$Treat2),
                      paste(btv.df$Treat2, " - ", btv.df$Treat1))
btv.df$Significant <- ifelse(btv.df$p.value < 0.05, "Yes", "No") #identify sig. differences
btv.df <- btv.df %>% arrange(AbsValue) # order by the magnitude of the effect size (step 1)
levels <- btv.df$Label # step 2
btv.df$Label <- factor(btv.df$Label, levels=levels) #step 3
btv.df %>% select(Label, AbsValue, AbsCI.lwr, AbsCI.upr, Significant, p.value) #inspect
```

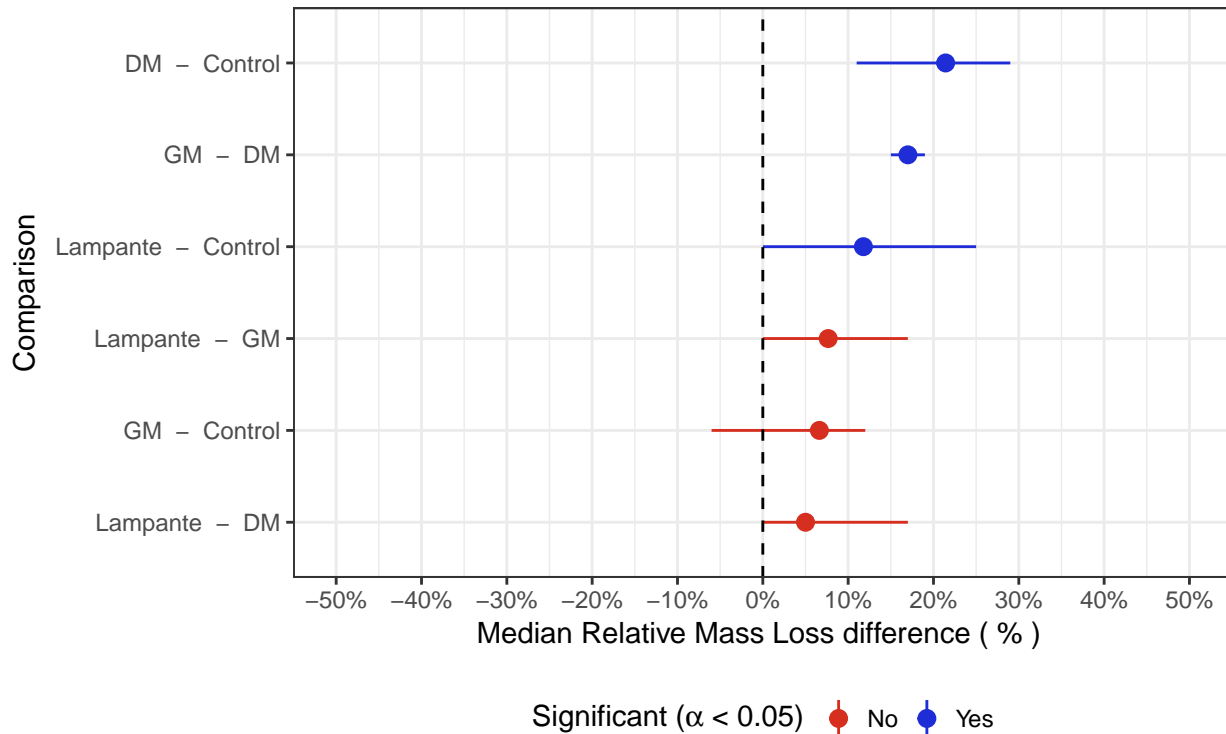
Label	AbsValue	AbsCI.lwr	AbsCI.upr	Significant	p.value
Lampante - DM	0.0500358	-0.0000308	0.1700008	No	0.0519241
GM - Control	0.0662661	-0.0600469	0.1200774	No	0.6480372
Lampante - GM	0.0766012	-0.0000625	0.1699508	No	0.0503150
Lampante - Control	0.1178695	0.0000450	0.2500215	Yes	0.0412435
GM - DM	0.1700422	0.1500060	0.1899541	Yes	0.0001668
DM - Control	0.2142257	0.1099611	0.2900207	Yes	0.0000675

```
ggplot(data=btv.df, aes(x=Label, y=AbsValue, ymin=AbsCI.lwr, ymax=AbsCI.upr,
                       colour=Significant)) + theme_bw() +
  geom_pointrange() +
  geom_hline(yintercept=0, linetype=2) +
  scale_colour_manual(values=c("#D62F1F", "#1E2DD6"),
                     name=expression(paste("Significant (", alpha, " < 0.05)")) +
  scale_y_continuous(labels=percent, limits=(c(-0.5, 0.5)), breaks=seq(-0.5, 0.5, 0.1)) +
  coord_flip() +
  labs(title="TV Inoculant, Beech",
       subtitle="Median difference between samples, with 95 % CIs",
       x="Comparison", y="Median Relative Mass Loss difference ( % ) ") +
```

```
theme(legend.position = "bottom")
```

## TV Inoculant, Beech

Median difference bewtween samples, with 95 % CIs



## Pine :: Inoculant GT

```
pgt <- fg.p16 %>% filter(Inoculant == "gt")
kruskal.test(pgt$RelativeMassLoss, as.factor(pgt$Treatment))
```

```
##
## Kruskal-Wallis rank sum test
##
## data: pgt$RelativeMassLoss and as.factor(pgt$Treatment)
## Kruskal-Wallis chi-squared = 16.818, df = 3, p-value = 0.0007704
```

The p-value suggests we need to check for differences between pairs of treatments.

```
pgt.df <- data_frame(Treat1=as.character("treat1"), Treat2=as.character("treat2"),
                    Loc.shift = 0, CI.lwr = 0, CI.upr = 0, p.value = 0)
pgt.c1 <- pgt %>% filter(Treatment %in% c("Control", "LO"))
pgt.cd <- pgt %>% filter(Treatment %in% c("Control", "DM"))
pgt.cg <- pgt %>% filter(Treatment %in% c("Control", "GM"))
pgt.ld <- pgt %>% filter(Treatment %in% c("LO", "DM"))
pgt.lg <- pgt %>% filter(Treatment %in% c("LO", "GM"))
pgt.dg <- pgt %>% filter(Treatment %in% c("DM", "GM"))
x <- wilcox.test(RelativeMassLoss~as.factor(Treatment), data=pgt.c1, conf.int=TRUE,
                conf.level=0.95, exact=TRUE)
```

```
## Warning in wilcox.test.default(x = c(0.06, 0.01, 0.09, 0.7, 0.01, 0.69, :
```

```

## cannot compute exact p-value with ties
## Warning in wilcox.test.default(x = c(0.06, 0.01, 0.09, 0.7, 0.01, 0.69, :
## cannot compute exact confidence intervals with ties
pgt.df <- bind_rows(pgt.df, data_frame(Treat1=as.character("Control"),
                                     Treat2=as.character("Lampante"),
                                     Loc.shift=x$estimate,
                                     CI.lwr=x$conf.int[1], CI.upr=x$conf.int[2],
                                     p.value=x$p.value))
x <- wilcox.test(RelativeMassLoss~as.factor(Treatment), data=pgt.cd, conf.int=TRUE,
                conf.level=0.95, exact=TRUE)

## Warning in wilcox.test.default(x = c(0.06, 0.01, 0.09, 0.7, 0.01, 0.69, :
## cannot compute exact p-value with ties
## Warning in wilcox.test.default(x = c(0.06, 0.01, 0.09, 0.7, 0.01, 0.69, :
## cannot compute exact confidence intervals with ties
pgt.df <- bind_rows(pgt.df, data_frame(Treat1=as.character("Control"),
                                     Treat2=as.character("DM"),
                                     Loc.shift=x$estimate,
                                     CI.lwr=x$conf.int[1], CI.upr=x$conf.int[2],
                                     p.value=x$p.value))
x <- wilcox.test(RelativeMassLoss~as.factor(Treatment), data=pgt.cg, conf.int=TRUE,
                conf.level=0.95, exact=TRUE)

## Warning in wilcox.test.default(x = c(0.06, 0.01, 0.09, 0.7, 0.01, 0.69, :
## cannot compute exact p-value with ties
## Warning in wilcox.test.default(x = c(0.06, 0.01, 0.09, 0.7, 0.01, 0.69, :
## cannot compute exact confidence intervals with ties
pgt.df <- bind_rows(pgt.df, data_frame(Treat1=as.character("Control"),
                                     Treat2=as.character("GM"),
                                     Loc.shift=x$estimate,
                                     CI.lwr=x$conf.int[1], CI.upr=x$conf.int[2],
                                     p.value=x$p.value))
x <- wilcox.test(RelativeMassLoss~as.factor(Treatment), data=pgt.ld, conf.int=TRUE,
                conf.level=0.95, exact=TRUE)

## Warning in wilcox.test.default(x = c(0.11, 0.09, 0.11, 0.12, 0.12, 0.14, :
## cannot compute exact p-value with ties
## Warning in wilcox.test.default(x = c(0.11, 0.09, 0.11, 0.12, 0.12, 0.14, :
## cannot compute exact confidence intervals with ties
pgt.df <- bind_rows(pgt.df, data_frame(Treat1=as.character("DM"),
                                     Treat2=as.character("Lampante"),
                                     Loc.shift=x$estimate,
                                     CI.lwr=x$conf.int[1], CI.upr=x$conf.int[2],
                                     p.value=x$p.value))
x <- wilcox.test(RelativeMassLoss~as.factor(Treatment), data=pgt.lg, conf.int=TRUE,
                conf.level=0.95, exact=TRUE)

## Warning in wilcox.test.default(x = c(0.26, 0.35, 0.31, 0.36, 0.32, 0.2, :
## cannot compute exact p-value with ties
## Warning in wilcox.test.default(x = c(0.26, 0.35, 0.31, 0.36, 0.32, 0.2, :
## cannot compute exact confidence intervals with ties

```

```

pgt.df <- bind_rows(pgt.df, data_frame(Treat1=as.character("GM"),
                                     Treat2=as.character("Lampante"),
                                     Loc.shift=x$estimate,
                                     CI.lwr=x$conf.int[1], CI.upr=x$conf.int[2],
                                     p.value=x$p.value))
x <- wilcox.test(RelativeMassLoss~as.factor(Treatment), data=pgt.dg, conf.int=TRUE,
                conf.level=0.95, exact=TRUE)

```

```

## Warning in wilcox.test.default(x = c(0.11, 0.09, 0.11, 0.12, 0.12, 0.14, :
## cannot compute exact p-value with ties

```

```

## Warning in wilcox.test.default(x = c(0.11, 0.09, 0.11, 0.12, 0.12, 0.14, :
## cannot compute exact confidence intervals with ties

```

```

pgt.df <- bind_rows(pgt.df, data_frame(Treat1=as.character("DM"),
                                     Treat2=as.character("GM"),
                                     Loc.shift=x$estimate,
                                     CI.lwr=x$conf.int[1], CI.upr=x$conf.int[2],
                                     p.value=x$p.value))

```

```
pgt.df <- pgt.df[-1,]
```

Examine the results and plot.

```

pgt.df$AbsValue <- abs(pgt.df$Loc.shift)
pgt.df$AbsCI.upr <- ifelse(pgt.df$Loc.shift < 0, pgt.df$CI.lwr * -1, pgt.df$CI.upr)
pgt.df$AbsCI.lwr <- ifelse(pgt.df$Loc.shift < 0, pgt.df$CI.upr * -1, pgt.df$CI.lwr)
# create a nice label for the chart
pgt.df$Label <- ifelse(pgt.df$Loc.shift > 1, paste(pgt.df$Treat1, " - ", pgt.df$Treat2),
                      paste(pgt.df$Treat2, " - ", pgt.df$Treat1))
pgt.df$Significant <- ifelse(pgt.df$p.value < 0.05, "Yes", "No") #identify sig. differences
pgt.df <- pgt.df %>% arrange(AbsValue) # order by the magnitude of the effect size (step 1)
levels <- pgt.df$Label # step 2
pgt.df$Label <- factor(pgt.df$Label, levels=levels) #step 3
pgt.df %>% select(Label, AbsValue, AbsCI.lwr, AbsCI.upr, Significant, p.value) #inspect

```

Label	AbsValue	AbsCI.lwr	AbsCI.upr	Significant	p.value
DM - Control	0.0199614	-0.0499958	0.4399599	No	0.5789196
Lampante - DM	0.0400765	0.0200844	0.0699530	Yes	0.0054694
Lampante - Control	0.0600542	-0.0000635	0.4599586	No	0.0544645
GM - Control	0.1035324	-0.2699507	0.1999798	No	0.2117280
GM - DM	0.1499362	0.0799839	0.2099828	Yes	0.0002632
Lampante - GM	0.1900067	0.1199795	0.2499527	Yes	0.0001766

```

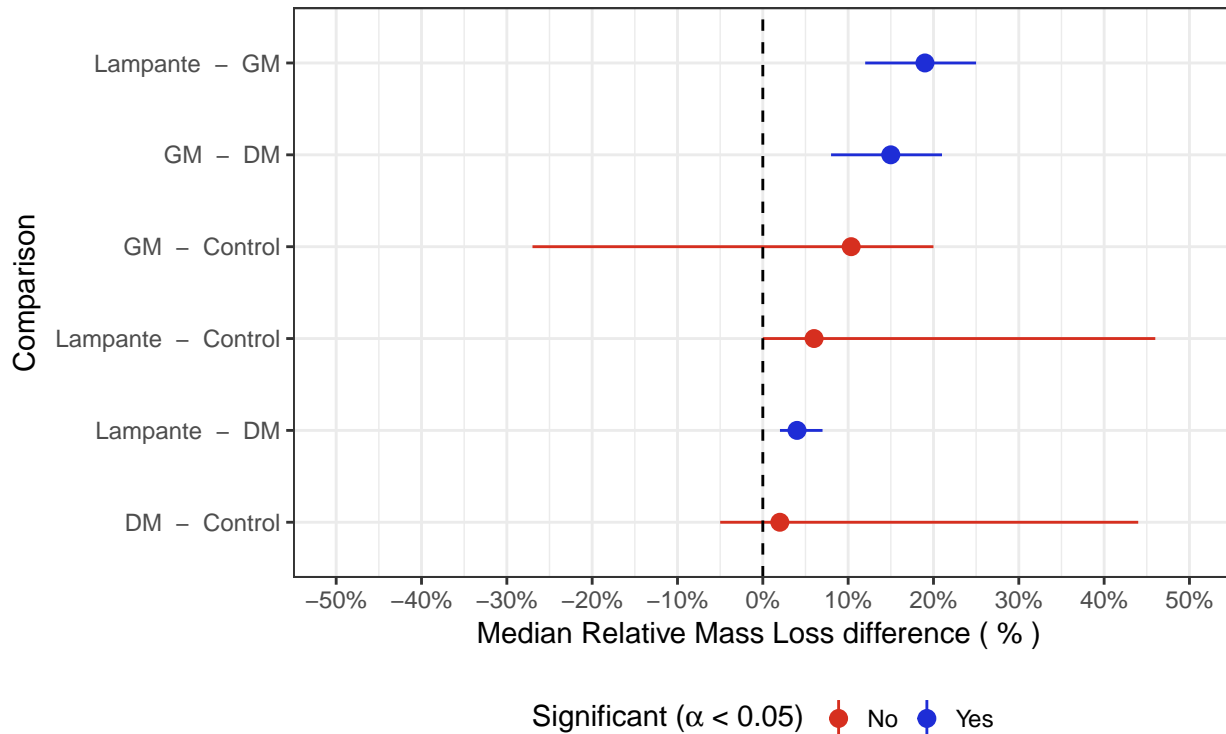
ggplot(data=pgt.df, aes(x=Label, y=AbsValue, ymin=AbsCI.lwr, ymax=AbsCI.upr,
                       colour=Significant)) + theme_bw() +
  geom_pointrange() +
  geom_hline(yintercept=0, linetype=2) +
  scale_colour_manual(values=c("#D62F1F", "#1E2DD6"),
                     name=expression(paste("Significant (", alpha, " < 0.05)")) +
  scale_y_continuous(labels=percent, limits=(c(-0.5, 0.5)), breaks=seq(-0.5, 0.5, 0.1)) +
  coord_flip() +
  labs(title="GT Inoculant, Pine",
       subtitle="Median difference between samples, with 95 % CIs",
       x="Comparison", y="Median Relative Mass Loss difference ( % ) ") +

```

```
theme(legend.position = "bottom")
```

## GT Inoculant, Pine

Median difference bewtween samples, with 95 % CIs



## Pine :: Inoculant PM

```
ppm <- fg.p16 %>% filter(Inoculant == "pm")
kruskal.test(pgt$RelativeMassLoss, as.factor(pgt$Treatment))
```

```
##
## Kruskal-Wallis rank sum test
##
## data: pgt$RelativeMassLoss and as.factor(pgt$Treatment)
## Kruskal-Wallis chi-squared = 16.818, df = 3, p-value = 0.0007704
```

The p-value suggests we need to check for differences between pairs of treatments.

```
ppm.df <- data_frame(Treat1=as.character("treat1"), Treat2=as.character("treat2"),
                     Loc.shift = 0, CI.lwr = 0, CI.upr = 0, p.value = 0)
ppm.c1 <- ppm %>% filter(Treatment %in% c("Control", "LO"))
ppm.cd <- ppm %>% filter(Treatment %in% c("Control", "DM"))
ppm.cg <- ppm %>% filter(Treatment %in% c("Control", "GM"))
ppm.ld <- ppm %>% filter(Treatment %in% c("LO", "DM"))
ppm.lg <- ppm %>% filter(Treatment %in% c("LO", "GM"))
ppm.dg <- ppm %>% filter(Treatment %in% c("DM", "GM"))
x <- wilcox.test(RelativeMassLoss~as.factor(Treatment), data=ppm.c1, conf.int=TRUE,
                 conf.level=0.95, exact=TRUE)
```

```
## Warning in wilcox.test.default(x = c(0.18, 0.18, 0.16, 0.17, 0.12, 0.08, :
```

```

## cannot compute exact p-value with ties
## Warning in wilcox.test.default(x = c(0.18, 0.18, 0.16, 0.17, 0.12, 0.08, :
## cannot compute exact confidence intervals with ties
ppm.df <- bind_rows(ppm.df, data_frame(Treat1=as.character("Control"),
                                       Treat2=as.character("Lampante"),
                                       Loc.shift=x$estimate,
                                       CI.lwr=x$conf.int[1], CI.upr=x$conf.int[2],
                                       p.value=x$p.value))
x <- wilcox.test(RelativeMassLoss~as.factor(Treatment), data=ppm.cd, conf.int=TRUE,
                conf.level=0.95, exact=TRUE)

## Warning in wilcox.test.default(x = c(0.18, 0.18, 0.16, 0.17, 0.12, 0.08, :
## cannot compute exact p-value with ties
## Warning in wilcox.test.default(x = c(0.18, 0.18, 0.16, 0.17, 0.12, 0.08, :
## cannot compute exact confidence intervals with ties
ppm.df <- bind_rows(ppm.df, data_frame(Treat1=as.character("Control"),
                                       Treat2=as.character("DM"),
                                       Loc.shift=x$estimate,
                                       CI.lwr=x$conf.int[1], CI.upr=x$conf.int[2],
                                       p.value=x$p.value))
x <- wilcox.test(RelativeMassLoss~as.factor(Treatment), data=ppm.cg, conf.int=TRUE,
                conf.level=0.95, exact=TRUE)

## Warning in wilcox.test.default(x = c(0.18, 0.18, 0.16, 0.17, 0.12, 0.08, :
## cannot compute exact p-value with ties
## Warning in wilcox.test.default(x = c(0.18, 0.18, 0.16, 0.17, 0.12, 0.08, :
## cannot compute exact confidence intervals with ties
ppm.df <- bind_rows(ppm.df, data_frame(Treat1=as.character("Control"),
                                       Treat2=as.character("GM"),
                                       Loc.shift=x$estimate,
                                       CI.lwr=x$conf.int[1], CI.upr=x$conf.int[2],
                                       p.value=x$p.value))
x <- wilcox.test(RelativeMassLoss~as.factor(Treatment), data=ppm.ld, conf.int=TRUE,
                conf.level=0.95, exact=TRUE)

## Warning in wilcox.test.default(x = c(0.13, 0.15, 0.04, 0.1, 0.1, 0.09,
## 0.08, : cannot compute exact p-value with ties
## Warning in wilcox.test.default(x = c(0.13, 0.15, 0.04, 0.1, 0.1, 0.09,
## 0.08, : cannot compute exact confidence intervals with ties
ppm.df <- bind_rows(ppm.df, data_frame(Treat1=as.character("DM"),
                                       Treat2=as.character("Lampante"),
                                       Loc.shift=x$estimate,
                                       CI.lwr=x$conf.int[1], CI.upr=x$conf.int[2],
                                       p.value=x$p.value))
x <- wilcox.test(RelativeMassLoss~as.factor(Treatment), data=ppm.lg, conf.int=TRUE,
                conf.level=0.95, exact=TRUE)

## Warning in wilcox.test.default(x = c(0.13, 0.14, 0.24, 0.22, 0.24, 0.21, :
## cannot compute exact p-value with ties
## Warning in wilcox.test.default(x = c(0.13, 0.14, 0.24, 0.22, 0.24, 0.21, :
## cannot compute exact confidence intervals with ties

```



```
ppm.df <- bind_rows(ppm.df, data_frame(Treat1=as.character("GM"),
                                     Treat2=as.character("Lampante"),
                                     Loc.shift=x$estimate,
                                     CI.lwr=x$conf.int[1], CI.upr=x$conf.int[2],
                                     p.value=x$p.value))
x <- wilcox.test(RelativeMassLoss~as.factor(Treatment), data=ppm.dg, conf.int=TRUE,
                conf.level=0.95, exact=TRUE)
```

```
## Warning in wilcox.test.default(x = c(0.13, 0.15, 0.04, 0.1, 0.1, 0.09,
## 0.08, : cannot compute exact p-value with ties
```

```
## Warning in wilcox.test.default(x = c(0.13, 0.15, 0.04, 0.1, 0.1, 0.09,
## 0.08, : cannot compute exact confidence intervals with ties
```

```
ppm.df <- bind_rows(ppm.df, data_frame(Treat1=as.character("DM"),
                                     Treat2=as.character("GM"),
                                     Loc.shift=x$estimate,
                                     CI.lwr=x$conf.int[1], CI.upr=x$conf.int[2],
                                     p.value=x$p.value))
```

```
ppm.df <- ppm.df[-1,]
```

Examine the results and plot.

```
ppm.df$AbsValue <- abs(ppm.df$Loc.shift)
ppm.df$AbsCI.upr <- ifelse(ppm.df$Loc.shift < 0, ppm.df$CI.lwr * -1, ppm.df$CI.upr)
ppm.df$AbsCI.lwr <- ifelse(ppm.df$Loc.shift < 0, ppm.df$CI.upr * -1, ppm.df$CI.lwr)
ppm.df$Label <- ifelse(ppm.df$Loc.shift > 1, paste(ppm.df$Treat1, " - ", ppm.df$Treat2),
                      paste(ppm.df$Treat2, " - ", ppm.df$Treat1))
ppm.df$Significant <- ifelse(ppm.df$p.value < 0.05, "Yes", "No") #identify sig. differences
ppm.df <- ppm.df %>% arrange(AbsValue) # order by the magnitude of the effect size (step 1)
levels <- ppm.df$Label # step 2
ppm.df$Label <- factor(ppm.df$Label, levels=levels) #step 3
ppm.df %>% select(Label, AbsValue, AbsCI.lwr, AbsCI.upr, Significant, p.value) #inspect
```

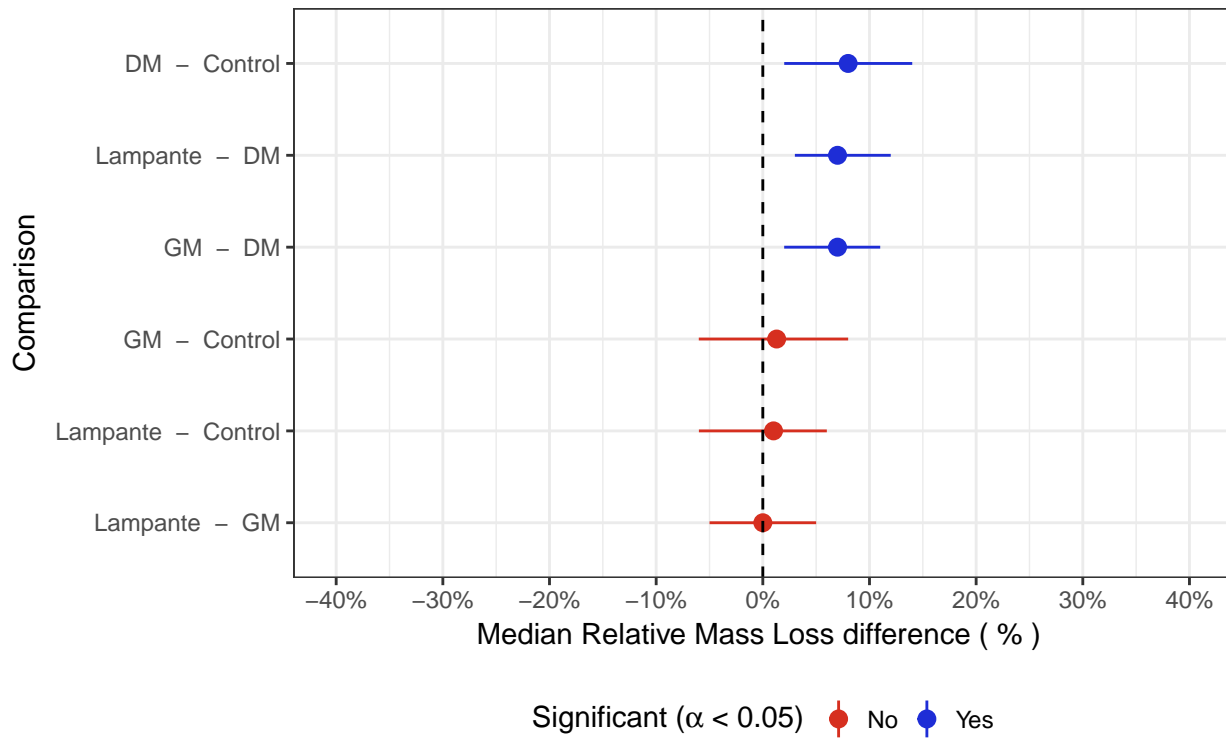
Label	AbsValue	AbsCI.lwr	AbsCI.upr	Significant	p.value
Lampante - GM	0.0000155	-0.0499983	0.0500308	No	0.8185046
Lampante - Control	0.0100256	-0.0599324	0.0600058	No	0.8286442
GM - Control	0.0129648	-0.0600513	0.0799896	No	0.8663136
GM - DM	0.0700354	0.0200071	0.1100638	Yes	0.0033629
Lampante - DM	0.0700461	0.0300559	0.1199314	Yes	0.0008078
DM - Control	0.0799810	0.0200683	0.1400909	Yes	0.0184493

```
ggplot(data=ppm.df, aes(x=Label, y=AbsValue, ymin=AbsCI.lwr, ymax=AbsCI.upr,
                       colour=Significant)) + theme_bw() +
  geom_pointrange() +
  geom_hline(yintercept=0, linetype=2) +
  scale_colour_manual(values=c("#D62F1F", "#1E2DD6"),
                     name=expression(paste("Significant (", alpha, " < 0.05)")) +
  scale_y_continuous(labels=percent, limits=(c(-0.4, 0.4)), breaks=seq(-0.4, 0.4, 0.1)) +
  coord_flip() +
  labs(title="PM Inoculant, Pine",
       subtitle="Median difference between samples, with 95 % CIs",
       x="Comparison", y="Median Relative Mass Loss difference ( % ) ") +
  theme(legend.position = "bottom")
```



## PM Inoculant, Pine

Median difference between samples, with 95 % CIs

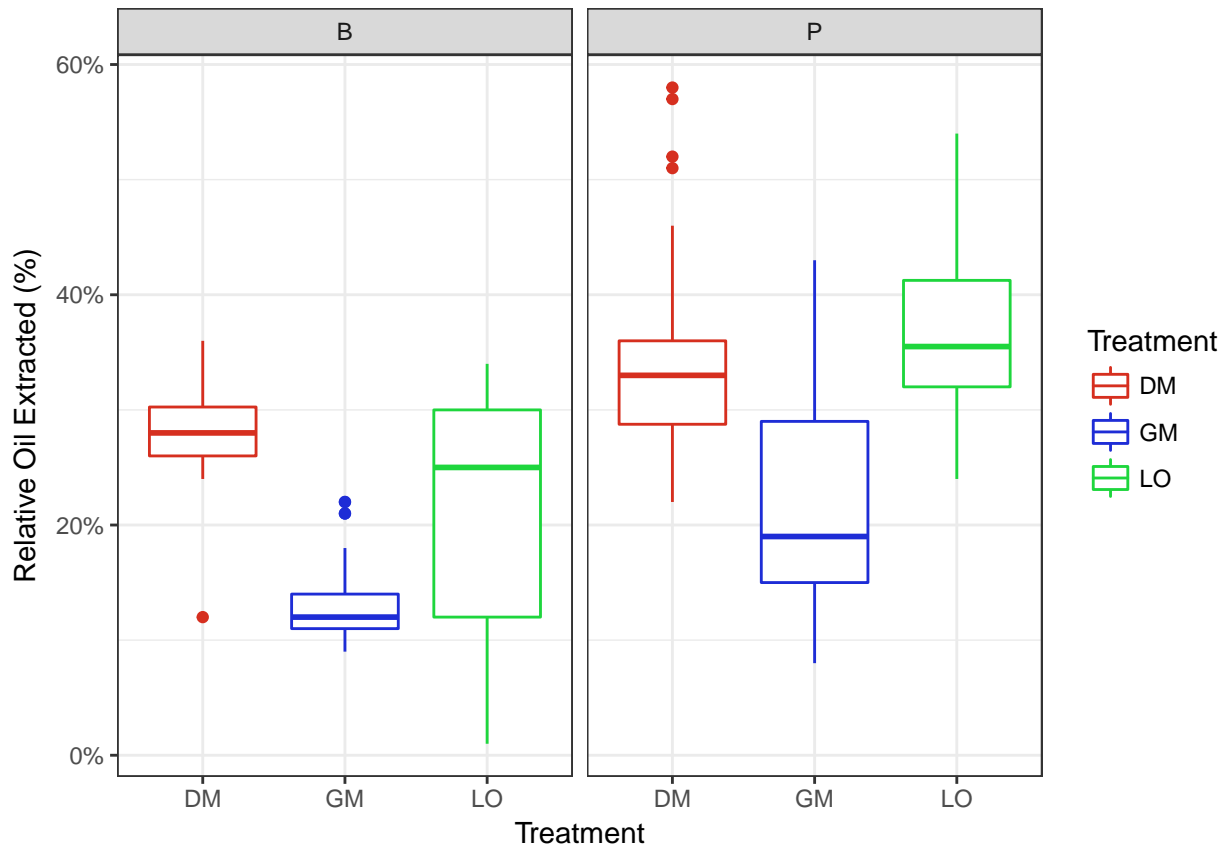


## Oil Extraction

Here we examine the amount of oil extracted by solvent extraction.

```
oe <- fg %>% filter(!SpecimenID %in% c("B-kontrola", "P-kontrola"))

ggplot(data=oe, aes(x=Treatment, y=OilExtract_rel, colour=Treatment)) + theme_bw() +
  geom_boxplot() +
  scale_colour_manual(values=c("#D62F1F", "#1E2DD6", "#1ED63D")) +
  facet_grid(~Species) +
  scale_y_continuous(labels=percent) +
  labs(y = "Relative Oil Extracted (%)")
```



## Fungal block test comparisons

For each species we compare the results of the fungal block tests by treatment.

### Beech

```
fbf <- oe %>% filter(Species == "B")
kruskal.test(fbf$OilExtract_rel, as.factor(fbf$Treatment))
```

```
##
## Kruskal-Wallis rank sum test
##
## data: fbf$OilExtract_rel and as.factor(fbf$Treatment)
## Kruskal-Wallis chi-squared = 51.852, df = 2, p-value = 5.501e-12
```

The Kruskal-Wallis test suggests a difference between some of the treatments. Now we need to compare each treatment individually. The following tests produce warnings because of ties, so exact p-values are not reported. (Warnings squelched).

```
fbf.df <- data_frame(Treat1=as.character("treat1"), Treat2=as.character("treat2"),
                    Loc.shift = 0, CI.lwr = 0, CI.upr = 0, p.value = 0)
fbf.ld <- fbf %>% filter(Treatment %in% c("LO", "DM"))
fbf.lg <- fbf %>% filter(Treatment %in% c("LO", "GM"))
fbf.dg <- fbf %>% filter(Treatment %in% c("DM", "GM"))

x <- wilcox.test(OilExtract_rel~as.factor(Treatment), data=fbf.ld, conf.int=TRUE,
```

```

      conf.level=0.95)
fbb.df <- bind_rows(fbb.df, data_frame(Treat1=as.character("DM"),
      Treat2=as.character("Lampante"),
      Loc.shift=x$estimate,
      CI.lwr=x$conf.int[1], CI.upr=x$conf.int[2],
      p.value=x$p.value))
x <- wilcox.test(OilExtract_rel~as.factor(Treatment), data=fbb.lg, conf.int=TRUE,
      conf.level=0.95)
fbb.df <- bind_rows(fbb.df, data_frame(Treat1=as.character("GM"),
      Treat2=as.character("Lampante"),
      Loc.shift=x$estimate,
      CI.lwr=x$conf.int[1], CI.upr=x$conf.int[2],
      p.value=x$p.value))
x <- wilcox.test(OilExtract_rel~as.factor(Treatment), data=fbb.dg, conf.int=TRUE,
      conf.level=0.95)
fbb.df <- bind_rows(fbb.df, data_frame(Treat1=as.character("DM"),
      Treat2=as.character("GM"),
      Loc.shift=x$estimate,
      CI.lwr=x$conf.int[1], CI.upr=x$conf.int[2],
      p.value=x$p.value))

fbb.df <- fbb.df[-1,]
fbb.df

```

Treat1	Treat2	Loc.shift	CI.lwr	CI.upr	p.value
DM	Lampante	0.0399818	0.0100378	0.0700167	0.0044855
GM	Lampante	-0.1200086	-0.1400314	-0.0800114	0.0002604
DM	GM	0.1599727	0.1400639	0.1700776	0.0000000

```

fbb.df$AbsValue <- abs(fbb.df$Loc.shift)
fbb.df$AbsCI.upr <- ifelse(fbb.df$Loc.shift < 0, fbb.df$CI.lwr * -1, fbb.df$CI.upr)
fbb.df$AbsCI.lwr <- ifelse(fbb.df$Loc.shift < 0, fbb.df$CI.upr * -1, fbb.df$CI.lwr)
fbb.df$Label <- ifelse(fbb.df$Loc.shift > 1, paste(fbb.df$Treat1, " - ", fbb.df$Treat2),
      paste(fbb.df$Treat2, " - ", fbb.df$Treat1))
fbb.df$Significant <- ifelse(fbb.df$p.value < 0.05, "Yes", "No") #identify sig. differences
fbb.df <- fbb.df %>% arrange(AbsValue) # order by the magnitude of the effect size (step 1)
levels <- fbb.df$Label # step 2
fbb.df$Label <- factor(fbb.df$Label, levels=levels) #step 3
fbb.df %>% select(Label, AbsValue, AbsCI.lwr, AbsCI.upr, Significant, p.value) #inspect

```

Label	AbsValue	AbsCI.lwr	AbsCI.upr	Significant	p.value
Lampante - DM	0.0399818	0.0100378	0.0700167	Yes	0.0044855
Lampante - GM	0.1200086	0.0800114	0.1400314	Yes	0.0002604
GM - DM	0.1599727	0.1400639	0.1700776	Yes	0.0000000

```

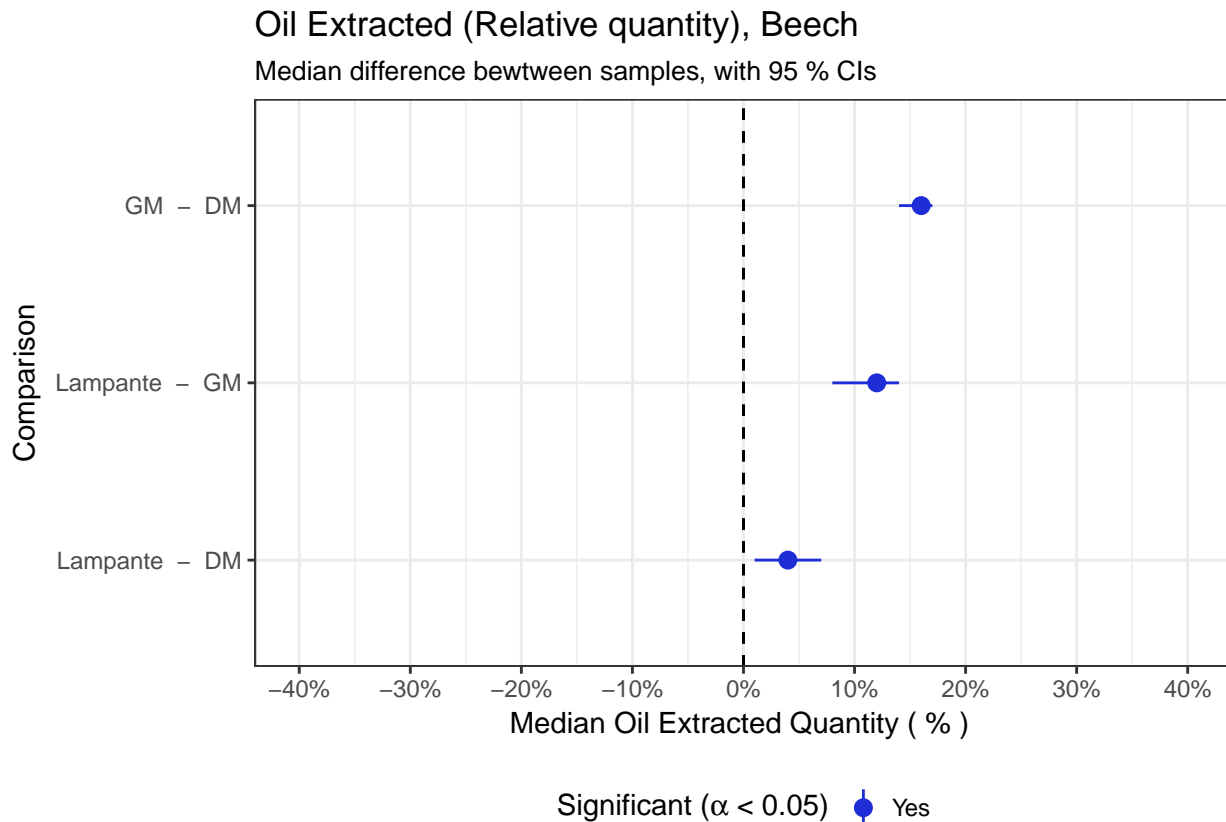
ggplot(data=fbb.df, aes(x=Label, y=AbsValue, ymin=AbsCI.lwr, ymax=AbsCI.upr,
      colour=Significant)) + theme_bw() +
  geom_pointrange() +
  geom_hline(yintercept=0, linetype=2) +
  scale_colour_manual(values=c("#1E2DD6"),
      name=expression(paste("Significant (", alpha, " < 0.05)")))) +

```

```

scale_y_continuous(labels=percent, limits=(c(-0.4, 0.4)), breaks=seq(-0.4, 0.4, 0.1)) +
geom_hline(yintercept=0, linetype=2) +
coord_flip() +
labs(title="Oil Extracted (Relative quantity), Beech",
      subtitle="Median difference between samples, with 95 % CIs",
      x="Comparison", y="Median Oil Extracted Quantity ( % ) ") +
theme(legend.position = "bottom")

```



## Pine

```

fbp <- oe %>% filter(Species == "P")
kruskal.test(fbp$OilExtract_rel, as.factor(fbp$Treatment))

```

```

##
## Kruskal-Wallis rank sum test
##
## data: fbp$OilExtract_rel and as.factor(fbp$Treatment)
## Kruskal-Wallis chi-squared = 41.031, df = 2, p-value = 1.231e-09

```

The Kruskal-Wallis test suggests a difference between some of the treatments. Now we need to compare each treatment individually. The following tests produce warnings because of ties, so exact p-values are not reported. (Warnings squelched).

```

fbp.df <- data_frame(Treat1=as.character("treat1"), Treat2=as.character("treat2"),
                    Loc.shift = 0, CI.lwr = 0, CI.upr = 0, p.value = 0)
fbp.ld <- fbp %>% filter(Treatment %in% c("LO", "DM"))
fbp.lg <- fbp %>% filter(Treatment %in% c("LO", "GM"))

```

```

fbp.dg <- fbp %>% filter(Treatment %in% c("DM", "GM"))

x <- wilcox.test(OilExtract_rel~as.factor(Treatment), data=fbp.ld, conf.int=TRUE,
                conf.level=0.95)
fbp.df <- bind_rows(fbp.df, data_frame(Treat1=as.character("DM"),
                                     Treat2=as.character("Lampante"),
                                     Loc.shift=x$estimate,
                                     CI.lwr=x$conf.int[1], CI.upr=x$conf.int[2],
                                     p.value=x$p.value))
x <- wilcox.test(OilExtract_rel~as.factor(Treatment), data=fbp.lg, conf.int=TRUE,
                conf.level=0.95)
fbp.df <- bind_rows(fbp.df, data_frame(Treat1=as.character("GM"),
                                     Treat2=as.character("Lampante"),
                                     Loc.shift=x$estimate,
                                     CI.lwr=x$conf.int[1], CI.upr=x$conf.int[2],
                                     p.value=x$p.value))
x <- wilcox.test(OilExtract_rel~as.factor(Treatment), data=fbp.dg, conf.int=TRUE,
                conf.level=0.95)
fbp.df <- bind_rows(fbp.df, data_frame(Treat1=as.character("DM"),
                                     Treat2=as.character("GM"),
                                     Loc.shift=x$estimate,
                                     CI.lwr=x$conf.int[1], CI.upr=x$conf.int[2],
                                     p.value=x$p.value))

fbp.df <- fbp.df[-1,]
fbp.df

```

Treat1	Treat2	Loc.shift	CI.lwr	CI.upr	p.value
DM	Lampante	-0.0299540	-0.0599400	-0.0000184	0.0372808
GM	Lampante	-0.1699894	-0.1999377	-0.1300802	0.0000000
DM	GM	0.1300330	0.0999561	0.1699968	0.0000005

```

fbp.df$AbsValue <- abs(fbp.df$Loc.shift)
fbp.df$AbsCI.upr <- ifelse(fbp.df$Loc.shift < 0, fbp.df$CI.lwr * -1, fbp.df$CI.upr)
fbp.df$AbsCI.lwr <- ifelse(fbp.df$Loc.shift < 0, fbp.df$CI.upr * -1, fbp.df$CI.lwr)
fbp.df$Label <- ifelse(fbp.df$Loc.shift > 1, paste(fbp.df$Treat1, " - ", fbp.df$Treat2),
                      paste(fbp.df$Treat2, " - ", fbp.df$Treat1))
fbp.df$Significant <- ifelse(fbp.df$p.value < 0.05, "Yes", "No") #identify sig. differences
fbp.df <- fbp.df %>% arrange(AbsValue) # order by the magnitude of the effect size (step 1)
levels <- fbp.df$Label # step 2
fbp.df$Label <- factor(fbp.df$Label, levels=levels) #step 3
fbp.df %>% select(Label, AbsValue, AbsCI.lwr, AbsCI.upr, Significant, p.value) #inspect

```

Label	AbsValue	AbsCI.lwr	AbsCI.upr	Significant	p.value
Lampante - DM	0.0299540	0.0000184	0.0599400	Yes	0.0372808
GM - DM	0.1300330	0.0999561	0.1699968	Yes	0.0000005
Lampante - GM	0.1699894	0.1300802	0.1999377	Yes	0.0000000

```

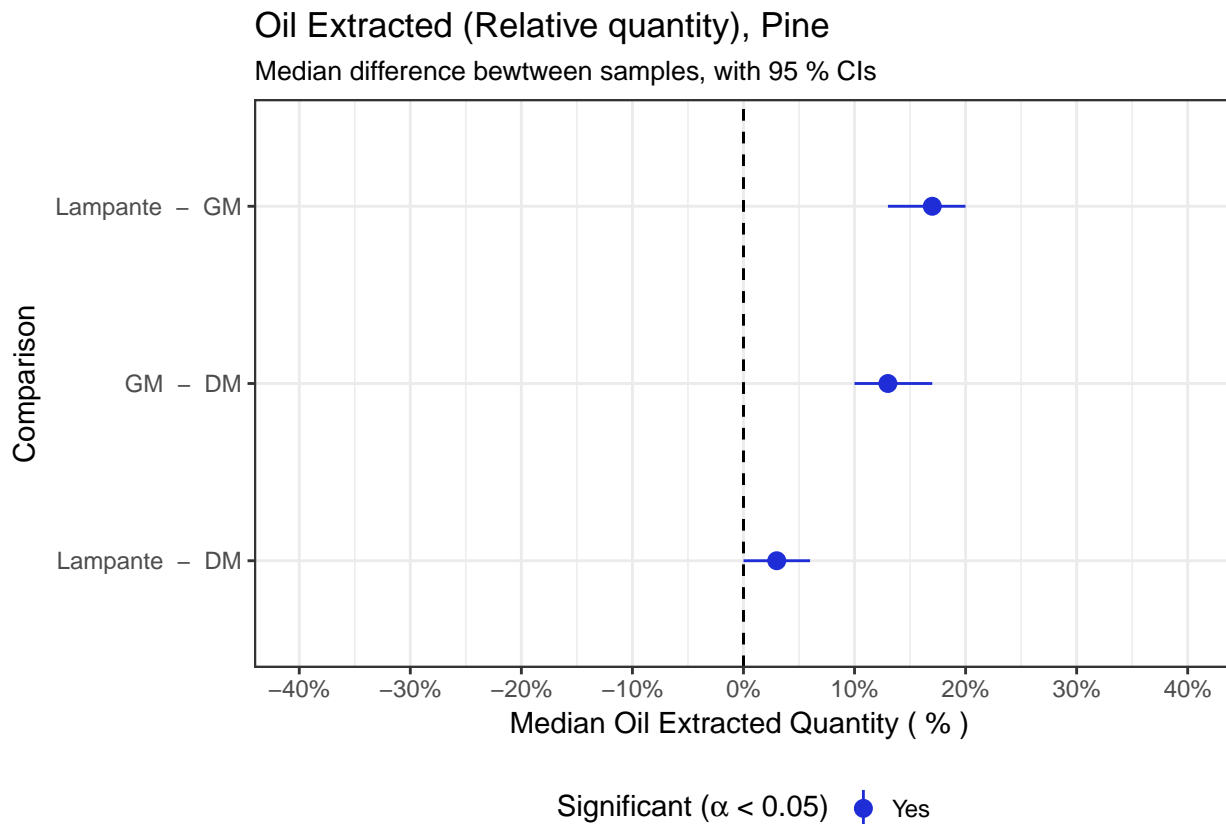
ggplot(data=fbp.df, aes(x=Label, y=AbsValue, ymin=AbsCI.lwr, ymax=AbsCI.upr,
                       colour=Significant)) + theme_bw() +
  geom_pointrange() +

```

```

geom_hline(yintercept=0, linetype=2) +
scale_colour_manual(values=c("#1E2DD6"),
                    name=expression(paste("Significant (", alpha, " < 0.05)"))) +
scale_y_continuous(labels=percent, limits=(c(-0.4, 0.4)), breaks=seq(-0.4, 0.4, 0.1)) +
geom_hline(yintercept=0, linetype=2) +
coord_flip() +
labs(title="Oil Extracted (Relative quantity), Pine",
      subtitle="Median difference between samples, with 95 % CIs",
      x="Comparison", y="Median Oil Extracted Quantity ( % ) ") +
theme(legend.position = "bottom")

```



## Acknowledgements

The authors graciously thank COST Actions FP1303 and FP1407, the IO-0035 Slovenian Infrastructural program for their funding and support and the European Commission for funding the InnoRenew CoE project (Grant Agreement #739574) under the Horizon2020 Widespread-Teaming program. The authors also thank and acknowledge the technical support of Primož Habjan, Courtney Williamson, and Marko Posavčević.

# Environment

```
sessionInfo()
```

```
## R version 3.4.3 (2017-11-30)
## Platform: x86_64-apple-darwin15.6.0 (64-bit)
## Running under: macOS High Sierra 10.13.3
##
## Matrix products: default
## BLAS: /Library/Frameworks/R.framework/Versions/3.4/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/3.4/Resources/lib/libRlapack.dylib
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] stats      graphics  grDevices  utils      datasets  methods    base
##
## other attached packages:
## [1] bindrcpp_0.2      ggalt_0.4.0      gridExtra_2.3    zoo_1.8-1
## [5] scales_0.5.0     forcats_0.2.0    stringr_1.2.0    dplyr_0.7.4
## [9] purrr_0.2.4      readr_1.1.1      tidyr_0.8.0      tibble_1.4.2
## [13] ggplot2_2.2.1    tidyverse_1.2.1
##
## loaded via a namespace (and not attached):
## [1] Rcpp_0.12.15      lubridate_1.7.2   lattice_0.20-35
## [4] assertthat_0.2.0 rprojroot_1.3-2   digest_0.6.15
## [7] proj4_1.0-8       psych_1.7.8       R6_2.2.2
## [10] cellranger_1.1.0 plyr_1.8.4        backports_1.1.2
## [13] evaluate_0.10.1  httr_1.3.1        highr_0.6
## [16] pillar_1.1.0     rlang_0.1.6       lazyeval_0.2.1
## [19] readxl_1.0.0     rstudioapi_0.7    extrafontdb_1.0
## [22] rmarkdown_1.8    labeling_0.3       extrafont_0.17
## [25] foreign_0.8-69   munsell_0.4.3     broom_0.4.3
## [28] compiler_3.4.3   modelr_0.1.1      pkgconfig_2.0.1
## [31] mnormt_1.5-5     htmltools_0.3.6   tidyselect_0.2.3
## [34] crayon_1.3.4     MASS_7.3-48       grid_3.4.3
## [37] nlme_3.1-131     jsonlite_1.5       Rttf2pt1_1.3.5
## [40] gtable_0.2.0     magrittr_1.5       KernSmooth_2.23-15
## [43] cli_1.0.0        stringi_1.1.6     reshape2_1.4.3
## [46] xml2_1.2.0       ash_1.0-15        RColorBrewer_1.1-2
## [49] tools_3.4.3      glue_1.2.0         maps_3.2.0
## [52] hms_0.4.1        parallel_3.4.3    yaml_2.1.16
## [55] colorspace_1.3-2 rvest_0.3.2       knitr_1.19
## [58] bindr_0.1         haven_1.1.1
```