

LTER Time Series Complete Dataset R Analysis

Cat Luria

July 13, 2015

Contents

1	Getting started	2
2	Alpha Diversity	2
2.1	Getting the table set up correctly	2
2.2	Richness	2
3	NMDS	7
3.1	Running NMDS with all samples	7
3.2	Add ellipses	11
4	Starting to Incorporate Ancillary Data	12
4.1	Data collected	12
4.2	envfit	13

1 Getting started

First, load all the necessary packages.

```
## Loading required package: tcltk
## Loading required package: vegan
## Loading required package: permute
## Loading required package: lattice
## This is vegan 2.3-0
## BiodiversityR 2.5-2: use function 'BiodiversityRGUI()' to launch the BiodiversityR Graphical User In
##
## gdata: read.xls support for 'XLS' (Excel 97-2004) files ENABLED.
##
## gdata: read.xls support for 'XLSX' (Excel 2007+) files ENABLED.
##
## Attaching package: 'gdata'
##
## The following object is masked from 'package:data.table':
##
##     last
##
## The following object is masked from 'package:stats':
##
##     nobs
##
## The following object is masked from 'package:utils':
##
##     object.size
```

Finally, I removed any OTUs identified as chloroplasts. This removed 0 OTUs from the OTU table.

2 Alpha Diversity

2.1 Getting the table set up correctly

2.2 Richness

Calculate diversity indices using BiodiversityR package. For this package, sites are rows and species are columns. Remember to exclude metadata columns.

```
#calculate richness by site
richness <- diversityresult(otutable_bac_samples_t_sorted,
                           index='richness', method='s')
#add metadata to richness
diversitytable <- cbind(meta_sorted, richness)
```

```

#Turn your 'treatment' column into a character vector
#diversitytable$Date <- as.character(diversitytable$Date)
#Then turn it back into an ordered factor
diversitytable$Date <- as.Date(diversitytable$Date, "%m/%d/%y")

#plotting richness for each sample
# p <-ggplot(diversitytable, aes(x=Date, y=richness)) +
#   geom_bar(stat="identity") +
#   theme_bw() +
#   theme(text = element_text(size=10),
#         axis.text.x = element_text(angle=90, vjust=1))
# p + facet_grid(.~Season, scales="free")

dummy_diversitytable <-
  read.table("dummy_diversitytable_2015aug25.txt", header=T, sep="\t")
dummy_diversitytable$Date <- as.Date(dummy_diversitytable$Date, "%m/%d/%y")

dummydiv <- rbind(diversitytable, dummy_diversitytable)

season_labeller <- function(var, value){
  value <- as.character(value)
  if (var=="Season") {
    value[value=="PAL0910"] <- "2009-2010"
    value[value=="PAL1011"] <- "2010-2011"
    value[value=="PAL1112"] <- "2011-2012"
    value[value=="PAL1213"] <- "2012-2013"
  }
  return(value)
}

dummysp <-ggplot(dummydiv, aes(x=Date, y=richness)) +
  geom_bar(stat="identity", width=3, color="#0072B2") +
  ylab("richness (# of OTUs)") +
  xlab("") +
  scale_x_date(breaks = date_breaks("months"),
    labels = date_format("%b")) +
  theme_bw() +
  theme(text = element_text(size=10),
        axis.text.x = element_text(vjust=1))

dummysp + facet_grid(.~Season, scales="free", labeller=season_labeller)

```

```
## Warning: position_stack requires non-overlapping x intervals
```

2.2.1 Compare across seasons

Surprisingly, ANOVA indicates that richness varies significantly between seasons. PAL1011 has significantly higher richness than PAL1112 and PAL1213.

```

aov.richness.season= aov(richness~Season,data=diversitytable)
summary(aov.richness.season)

```

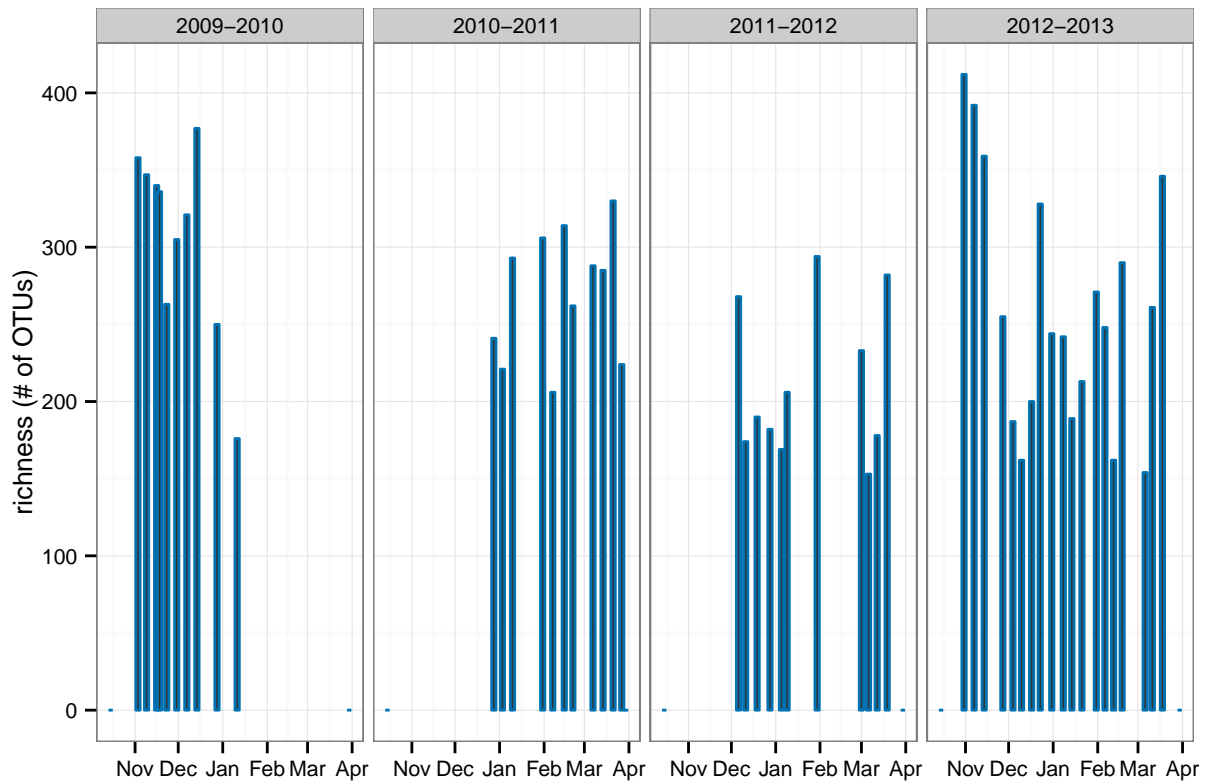


Figure 1: Observed OTUs per each individual library.

```
##           Df Sum Sq Mean Sq F value    Pr(>F)
## Season      3  49124  16374.8   4.1318 0.01112 *
## Residuals  47 186264   3963.1
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
tuk<- TukeyHSD(aov.richness.season)
tuk
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = richness ~ Season, data = diversitytable)
##
## $Season
##           diff           lwr           upr         p adj
## PAL1011-PAL0910 -37.300000 -110.559498   35.959498 0.53270050
## PAL1112-PAL0910 -95.572727 -168.832225  -22.313230 0.00591378
## PAL1213-PAL0910 -48.615789 -114.120545   16.888966 0.21132119
## PAL1112-PAL1011 -58.272727 -129.766676   13.221222 0.14639770
## PAL1213-PAL1011 -11.315789  -74.839830   52.208251 0.96434319
## PAL1213-PAL1112  46.956938  -16.567103  110.480978 0.21434960
```

```
#write a function to calculate SD
myFunc = function(x) {
```

```

result = c(mean(x) - sd(x), mean(x) + sd(x))
names(result) = c("ymin", "ymax")
result
}

#aggregate by month and apply SD function
aggregate(diversitytable$richness, by = list(diversitytable$Season), FUN = myFunc)

```

```

##   Group.1    x.ymin    x.ymax
## 1 PAL0910 246.22073 368.37927
## 2 PAL1011 228.25314 311.74686
## 3 PAL1112 162.17913 261.27541
## 4 PAL1213 180.25870 337.10973

```

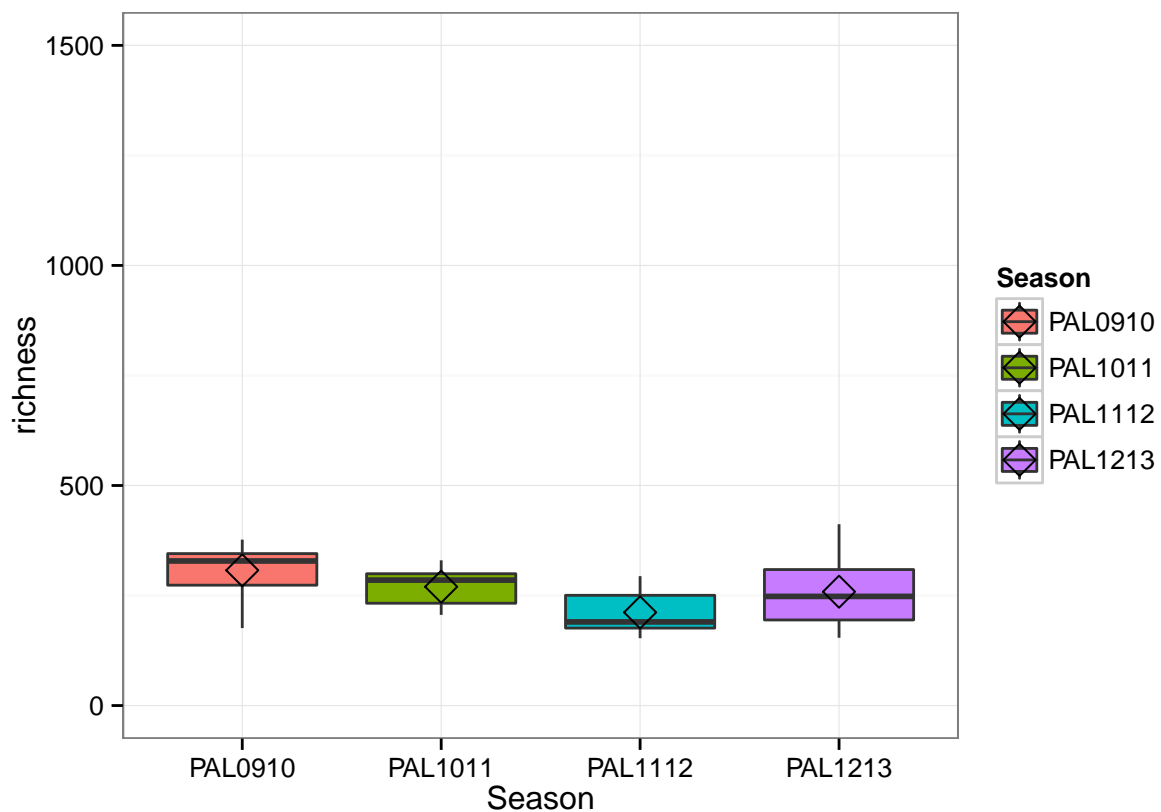


Figure 2: Mean richness (observed OTUs) for each PAL LTER season. Mean is indicated by diamond.

2.2.2 Compare across months

Richness does not vary between months.

```

##           Df Sum Sq Mean Sq F value    Pr(>F)
## Month      5  79740 15948.1   4.6108 0.00176 **
## Residuals 45 155648  3458.8
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

```
##   Group.1    x.ymin    x.ymax
## 1     Dec 172.80043 307.81496
## 2     Feb 191.37626 302.62374
## 3     Jan 184.84362 284.24729
## 4     Mar 182.07687 315.01404
## 5     Nov 282.77019 373.89648
## 6     Oct          NA         NA
```

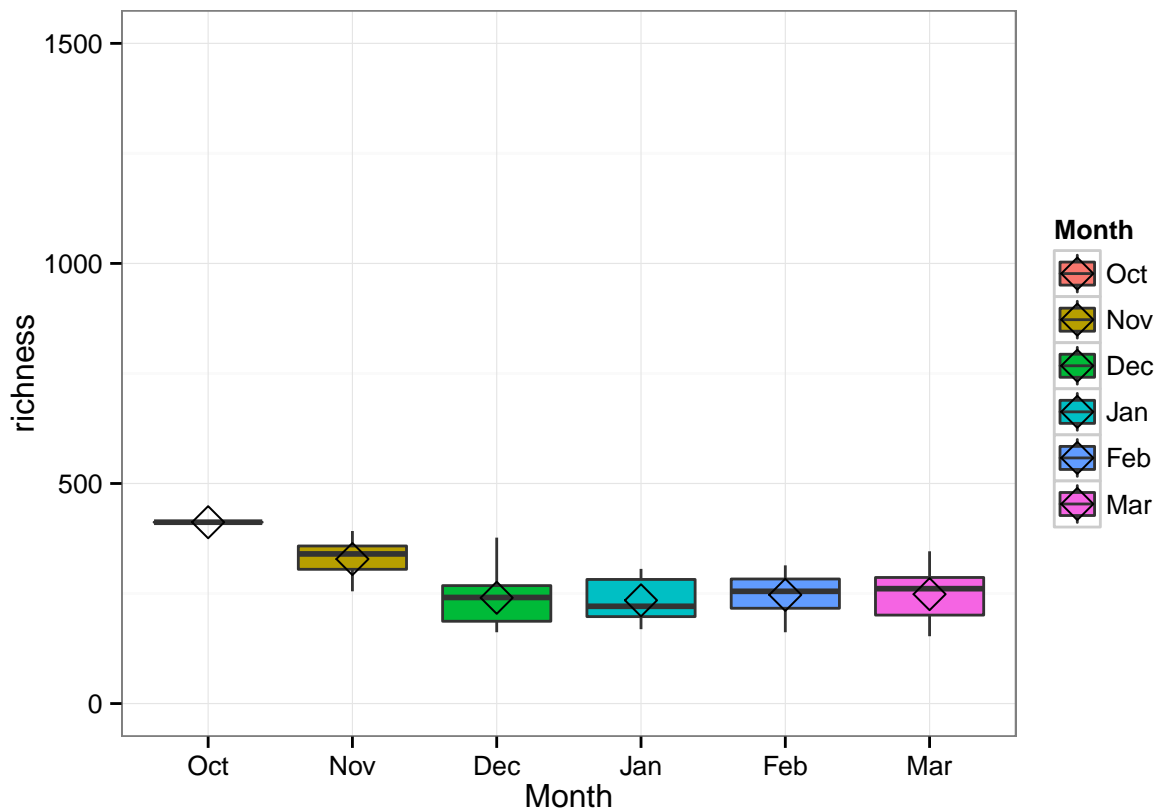


Figure 3: Mean richness (observed OTUs) for each month grouped across all seasons. Error bars are standard deviation.

```
lm_richness_daylength <- lm(richness~DayLength, data=diversitytable)
summary(lm_richness_daylength)
```

```
##
## Call:
## lm(formula = richness ~ DayLength, data = diversitytable)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -119.4630  -56.0513   -0.6589   49.3507  147.7555
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  310.9835    55.7780   5.5754 1.048e-06 ***
## DayLength    -2.7954     3.0440  -0.9183   0.3629
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 68.721 on 49 degrees of freedom
## Multiple R-squared:  0.016919,    Adjusted R-squared:  -0.0031436
## F-statistic: 0.84331 on 1 and 49 DF,  p-value: 0.36295
```

```
plot(richness~DayLength, data=diversitytable)
abline(lm_richness_daylength, col="red")
```

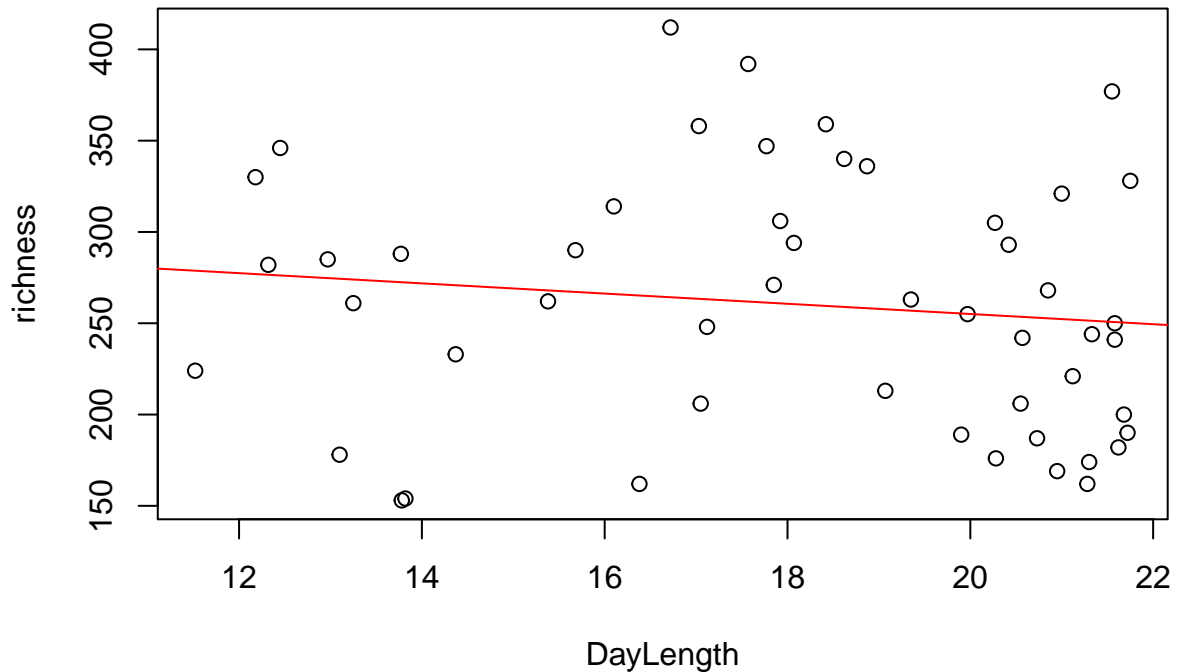


Figure 4: Richness (observed OTUs) versus day length (hours between sunrise and sunset) during summer months at Palmer Station.

Richness declines with increasing daylength but the relationship is weak ($p=0.08$).

3 NMDS

Run NMDS using vegan package. Then convert to resulting MDS coordinates to a data frame. Finally cbind map file to mds coordinate table.

3.1 Running NMDS with all samples

```
otutable_bac.mds <- metaMDS(otutable_bac_samples_t_sorted,
                             distance="bray", k=2, autotransform=FALSE)
```

```
## Run 0 stress 0.16537558
## Run 1 stress 0.16636297
```

```
## Run 2 stress 0.16818958
## Run 3 stress 0.20742313
## Run 4 stress 0.20664278
## Run 5 stress 0.16641047
## Run 6 stress 0.16813757
## Run 7 stress 0.16703354
## Run 8 stress 0.21893762
## Run 9 stress 0.16639177
## Run 10 stress 0.16818921
## Run 11 stress 0.16537557
## ... New best solution
## ... procrustes: rmse 2.5262598e-05  max resid 0.00010162358
## *** Solution reached
```

```
#convert resulting MDS coordinates to a data frame
otutable_bac.mds.points <- as.data.frame(otutable_bac.mds$points)
#add site codes and other factors to data frame
coded.otutable_bac.mds <- cbind(otutable_bac.mds.points, meta_sorted)

coded.otutable_bac.mds$Month <- as.factor(coded.otutable_bac.mds$Month)
coded.otutable_bac.mds$Month = ordered(coded.otutable_bac.mds$Month,
                                       levels=c("Oct", "Nov", "Dec", "Jan", "Feb", "Mar"))
```

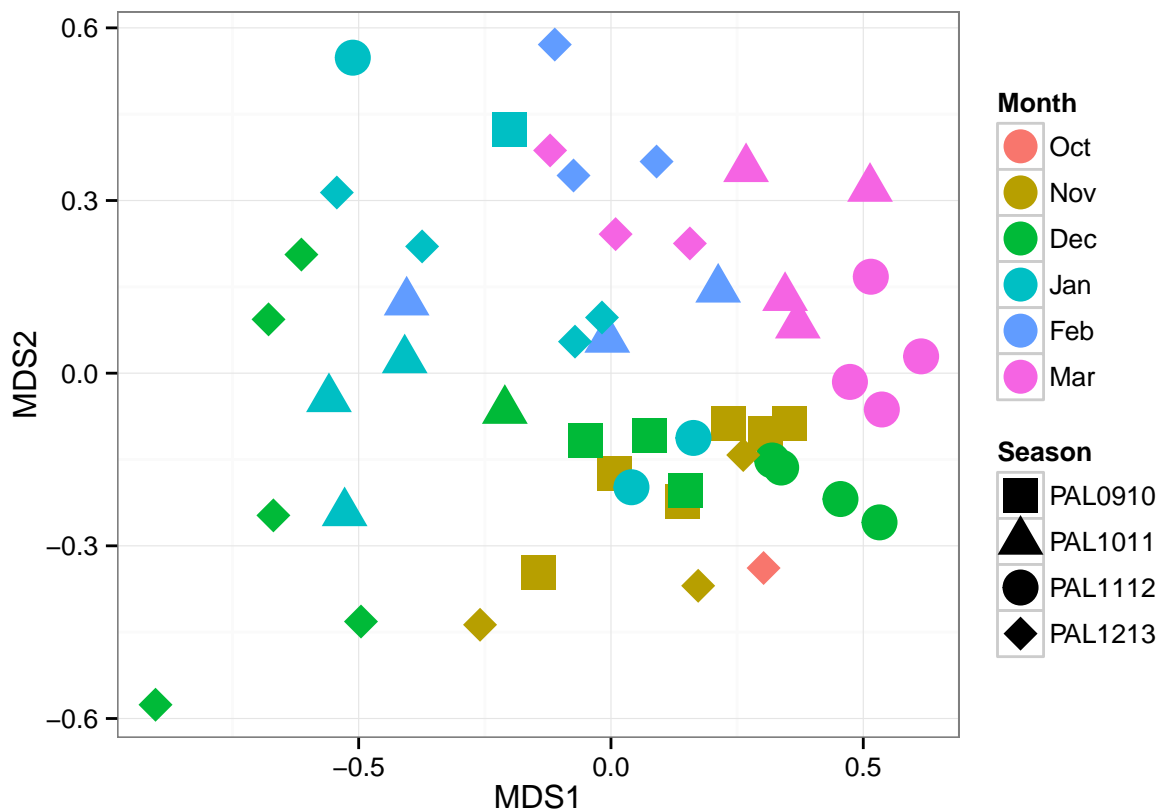


Figure 5: NMDS with all samples. Sample shape is coded by LTER season. Sample color is coded by month.

```
## Square root transformation
```

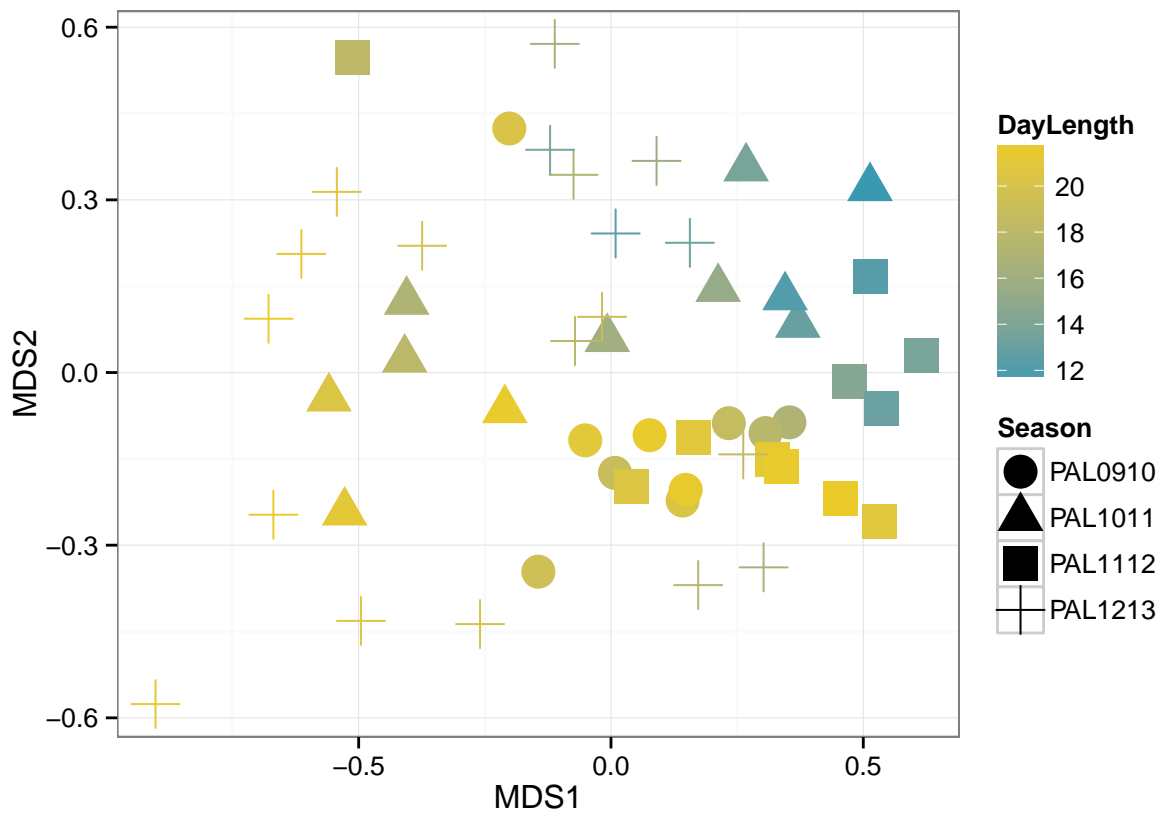



Figure 6: NMDS with all samples, results color-coded according to day length, i.e. hours between official sunrise and official sunset.

```

## Wisconsin double standardization
## Run 0 stress 0.22169081
## Run 1 stress 0.23768789
## Run 2 stress 0.2712744
## Run 3 stress 0.22256066
## Run 4 stress 0.22143307
## ... New best solution
## ... procrustes: rmse 0.01954813  max resid 0.11197043
## Run 5 stress 0.25196975
## Run 6 stress 0.2215754
## ... procrustes: rmse 0.0098100138  max resid 0.061595463
## Run 7 stress 0.22256075
## Run 8 stress 0.22205071
## Run 9 stress 0.22167754
## ... procrustes: rmse 0.019252151  max resid 0.11508775
## Run 10 stress 0.23079262
## Run 11 stress 0.23655676
## Run 12 stress 0.22153478
## ... procrustes: rmse 0.017476436  max resid 0.1138541
## Run 13 stress 0.22157584
## ... procrustes: rmse 0.0098256102  max resid 0.061864121
## Run 14 stress 0.24938023
## Run 15 stress 0.2215348
## ... procrustes: rmse 0.017493292  max resid 0.11403216
## Run 16 stress 0.2215754
## ... procrustes: rmse 0.0098065296  max resid 0.061525248
## Run 17 stress 0.23836937
## Run 18 stress 0.2500053
## Run 19 stress 0.27454376
## Run 20 stress 0.22157549
## ... procrustes: rmse 0.009817484  max resid 0.0617154

## Run 0 stress 0.14379387
## Run 1 stress 0.14988304
## Run 2 stress 0.15118095
## Run 3 stress 0.14379386
## ... New best solution
## ... procrustes: rmse 2.6148342e-05  max resid 5.84552e-05
## *** Solution reached

## Run 0 stress 0.10193732
## Run 1 stress 0.10193732
## ... procrustes: rmse 2.7734433e-06  max resid 1.0988249e-05
## *** Solution reached

## Too few points to calculate an ellipse
## Too few points to calculate an ellipse
## Too few points to calculate an ellipse
## Too few points to calculate an ellipse
## Too few points to calculate an ellipse

```

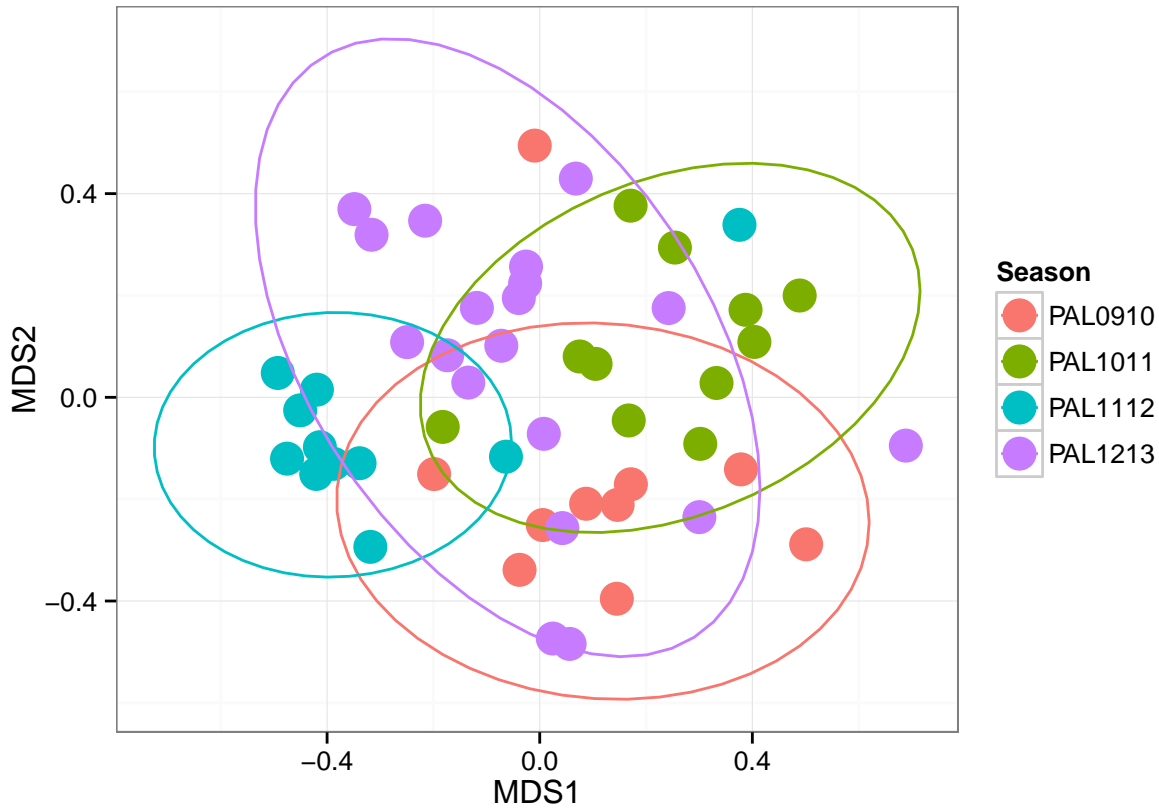


Figure 7: NMDS with all samples after Wisconsin double transformation.

```
## Run 0 stress 0.10193732
## Run 1 stress 0.10193732
## ... New best solution
## ... procrustes: rmse 2.8568924e-06  max resid 9.4519266e-06
## *** Solution reached
```

3.2 Add ellipses

```
## Run 0 stress 0.14379387
## Run 1 stress 0.15118101
## Run 2 stress 0.15118097
## Run 3 stress 0.141605
## ... New best solution
## ... procrustes: rmse 0.017075107  max resid 0.093428959
## Run 4 stress 0.141605
## ... procrustes: rmse 4.2897057e-06  max resid 2.0660522e-05
## *** Solution reached
```

```
##      NMDS1      NMDS2  Season
## 1 1.00009104 0.022991340 PAL1011
## 2 0.99815305 0.042281643 PAL1011
## 3 0.99234671 0.060806854 PAL1011
## 4 0.98269495 0.078493862 PAL1011
```

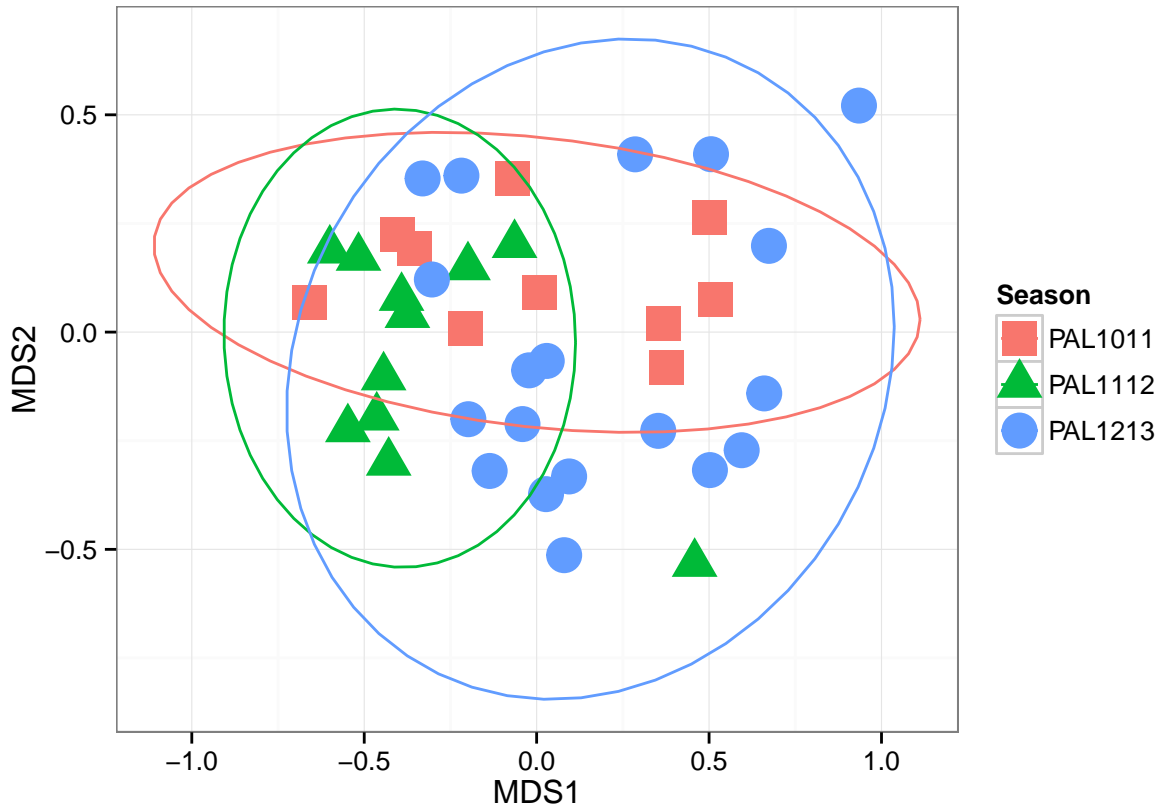


Figure 8: NMDS without PAL0910.

```
## 5 0.96923586 0.095272866 PAL1011
## 6 0.95202255 0.111077646 PAL1011
```

```
##      group      x      y
## 1 PAL1011  0.017968569 -0.15158212
## 2 PAL1112 -0.334136551  0.07178316
## 3 PAL1213  0.183990335  0.03822139
```

4 Starting to Incorporate Ancillary Data

4.1 Data collected

The following metadata has been collected so far. The remaining data are not yet on Datazoo. “X” indicates that the data has been compiled. “ND” indicates that the data does not exist.

Data	PAL0910	PAL1011	PAL1112	PAL1213	PAL1314
Temperature	x	x	x		
Salinity	x	x	x		
Abundance	x	x	x	x	x
Thymidine	x	ND	ND	ND	ND
Leucine	x	x	x	x	x
DOC	x	x	x		

Data	PAL0910	PAL1011	PAL1112	PAL1213	PAL1314
POC	x	x	x		
PON	x	x	x		
Phosphate	x	x	x	x	x
Silicate	x	x	x	x	x
NitriteNitrate	x	x	x	x	x
PrimaryProd	x	x	x	x	x
PrimProdSTD	x	x	x	x	x
Chlorophyll	x	x	x	x	x
Phaeopigment	x	x	x	x	x
HPLC					

4.2 envfit

```

#===not working===
# otutable_bac.mds <- metaMDS(otutable_bac_samples_t[-(1:3)],,
#                               distance="bray", k=2, autotransform=FALSE)
#
# meta_sorted_NA <- as.data.frame(meta_sorted_NA)
# ef <- envfit(otutable_bac.mds, meta_sorted[, c(15:32)], permu = 999)
#
# meta_sorted_NA_num <- meta_sorted_NA[,15:32]
# meta_sorted_NA_num <- meta_sorted_NA_num[,-(3:5)]
# meta_sorted_NA_num <- meta_sorted_NA_num[,-(6:8)]
# meta_sorted_NA_num <- as.data.frame(meta_sorted_NA_num, stringsAsFactors=FALSE)
#
# ef <- envfit(otutable_bac.mds, meta_sorted_NA_num, na.rm=TRUE)
# vf <- vectorfit(otutable_bac.mds, meta_sorted_NA, na.rm=TRUE)
#
# plot(otutable_bac.mds, display = "sites")
# plot(ef, p.max = 0.1)

```

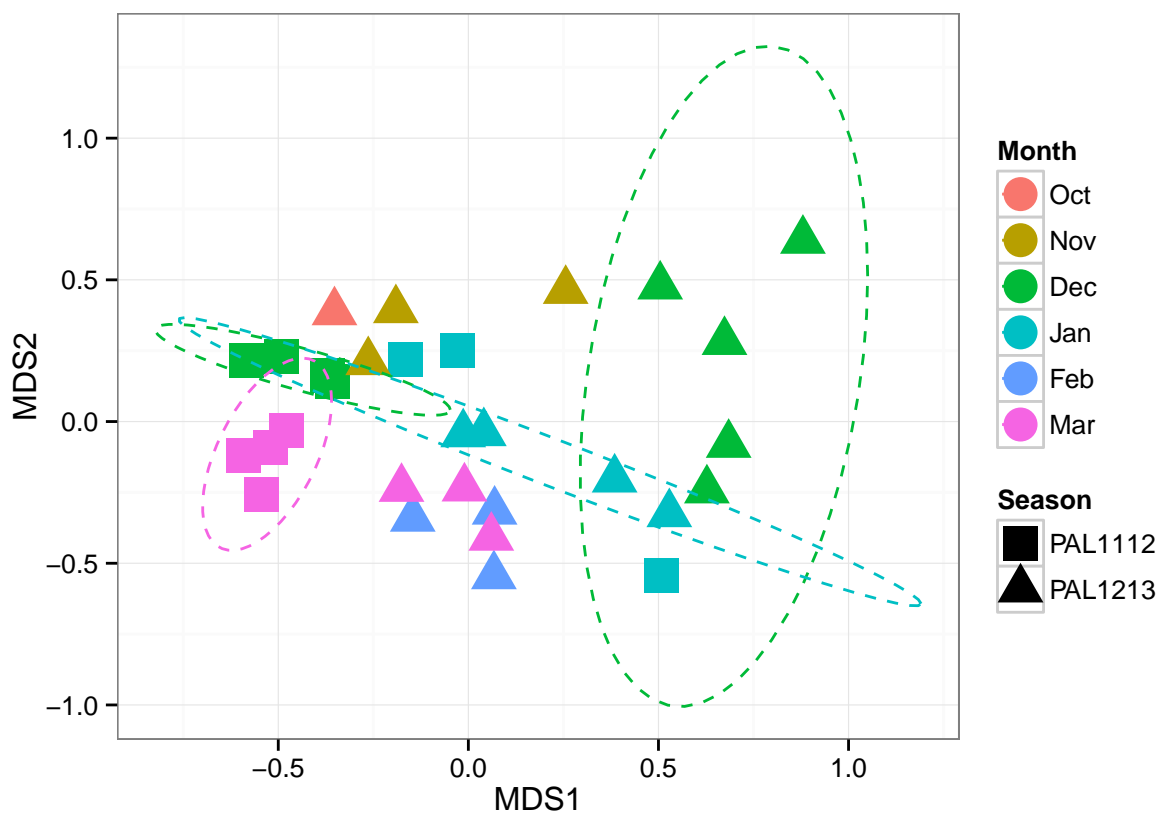


Figure 9: NMDS without PAL0910 or PAL1011.

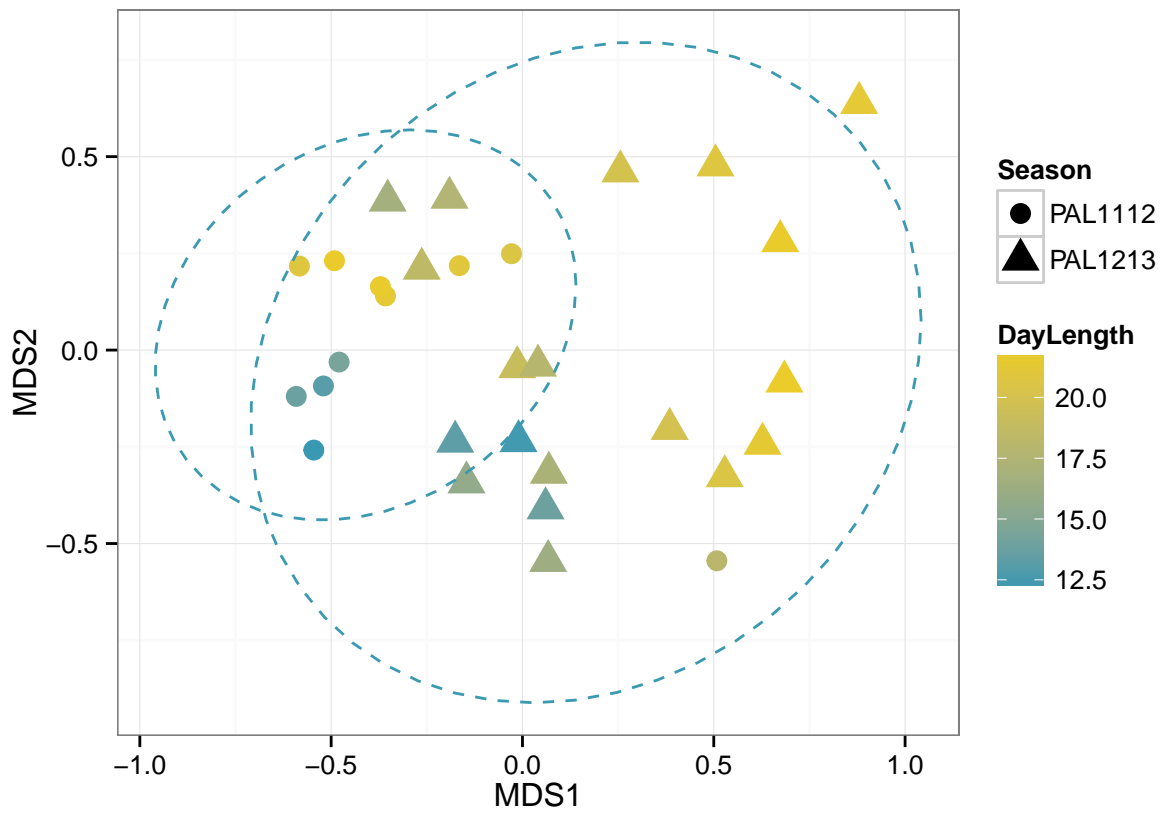


Figure 10: NMDS results without PAL0910 or PAL1011, samples color-coded according to day length, i.e. hours between official sunrise and official sunset.

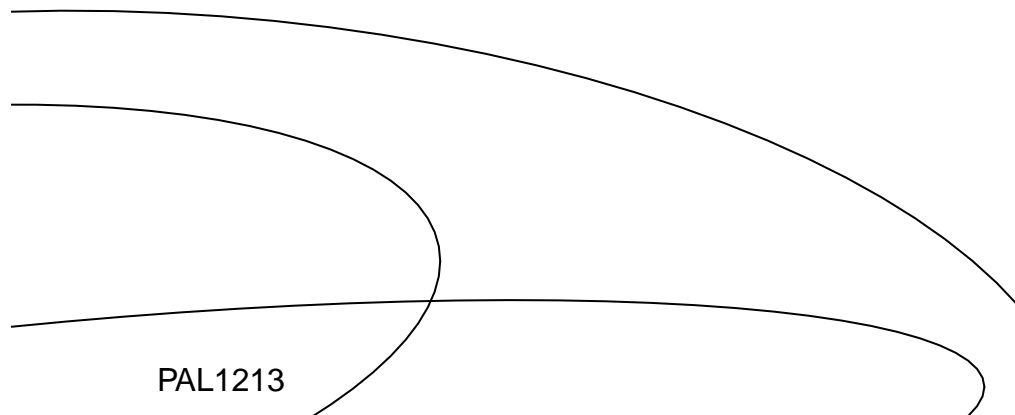


Figure 11: NMDS without PAL0910. Ellipses represent 95% confidence intervals.

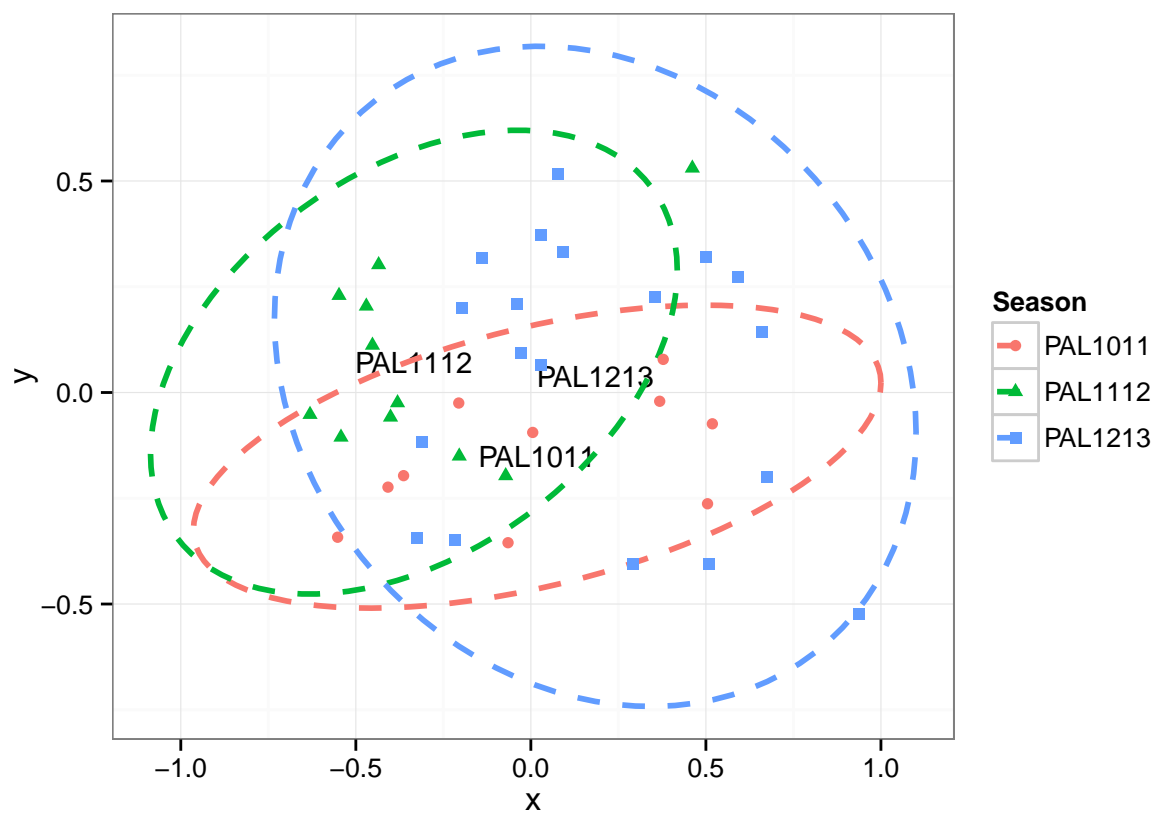


Figure 12: NMDS without PAL0910. Ellipses represent 95% confidence intervals.