

# Skin Microbiome

*Honors Thesis Student: Sarah Cummins; Advisor: Jay Lennon*

*04 April, 2015*

## 1) SETUP

### A. Retrieve and Set Your Working Directory

```
rm(list = ls())
getwd()
setwd("~/GitHub/skin")
```

### B. Load Packages

```
require("vegan")
```

```
## Loading required package: vegan
## Loading required package: permute
## Loading required package: lattice
## This is vegan 2.2-1
```

## 2) ANALYSIS OF DIFFERENT HABITATS

### A. Load data

```
# Load raw data
hab <- read.table("skin.habitat.txt", sep = "\t", header = TRUE)

# Look at raw data structure
str(hab)
```

```
## 'data.frame': 30 obs. of 6 variables:
## $ subject: Factor w/ 10 levels "C1","C10","C2",..: 1 1 1 3 3 3 4 4 4 5 ...
## $ habitat: Factor w/ 3 levels "arm","back","knee": 2 1 3 2 1 3 2 1 3 2 ...
## $ active : int 110 6 14 1777 6 5 30 6 11 323 ...
## $ dead : int 1095 20 52 4273 19 13 196 26 153 2776 ...
## $ dormant: int 406 102 403 297 56 94 72 115 185 880 ...
## $ total : int 1611 128 469 6347 81 112 298 147 349 3979 ...
```

### B. Calculate percentages of each metabolic class

```

# Percent active, dead, and dormanc
act.per <- hab$active/hab$total
dead.per <- hab$dead/hab$total
dorm.per <- hab$dormant/hab$total

# Make new dataframe with percent metabolic class data
hab.per <- data.frame(hab[1:2], act.per, dead.per, dorm.per)

```

### C. Make a summary table for metabolic classes by habitat

```

sem <- function(x){
  sd(x)/sqrt(length(x))
}

table <- aggregate(hab.per[, c("act.per", "dead.per", "dorm.per")],
  by = list(hab.per$habitat), function(x) c(mean=mean(x), sem=sem(x)))
colnames(table) <- c("habitat", "active", "dead", "dormant")

print(table)

```

```

##  habitat active.mean active.sem dead.mean dead.sem dormant.mean
## 1    arm      0.06220   0.01004   0.39263  0.05369      0.54517
## 2   back      0.11223   0.02567   0.66962  0.03496      0.21815
## 3   knee      0.08449   0.02015   0.44416  0.06236      0.47134
##  dormant.sem
## 1      0.05200
## 2      0.02920
## 3      0.06693

```

### D. Make stacked bar chart for each subject and differen habitats

```

back <- subset(hab.per, habitat == "back")
back.sub <- t(back[,c(5,4,3)])
colnames(back.sub) <- 1:10

arm <- subset(hab.per, habitat == "arm")
arm.sub <- t(arm[,c(5,4,3)])
colnames(arm.sub) <- 1:10

knee <- subset(hab.per, habitat == "knee")
knee.sub <- t(knee[,c(5,4,3)])
colnames(knee.sub) <- 1:10

par(mfrow = c(1,1), mar = c(1,7,2,5.5))
bar.layout <- layout(rbind(1, 2, 3, 4), height = c(4, 4, 5, 1))
#layout.show(bar.layout)

barplot.back <- barplot(as.matrix(back.sub),
  ylim = c(0, 1), las = 1,

```

```

    beside = FALSE,
    cex.axis = 1.5,
    axisnames = FALSE,
    col = c("darkblue", "red", "yellow"), plot = TRUE)
mtext("Back", side = 4, las = 1, cex = 1.5, adj = 0, line = -1)

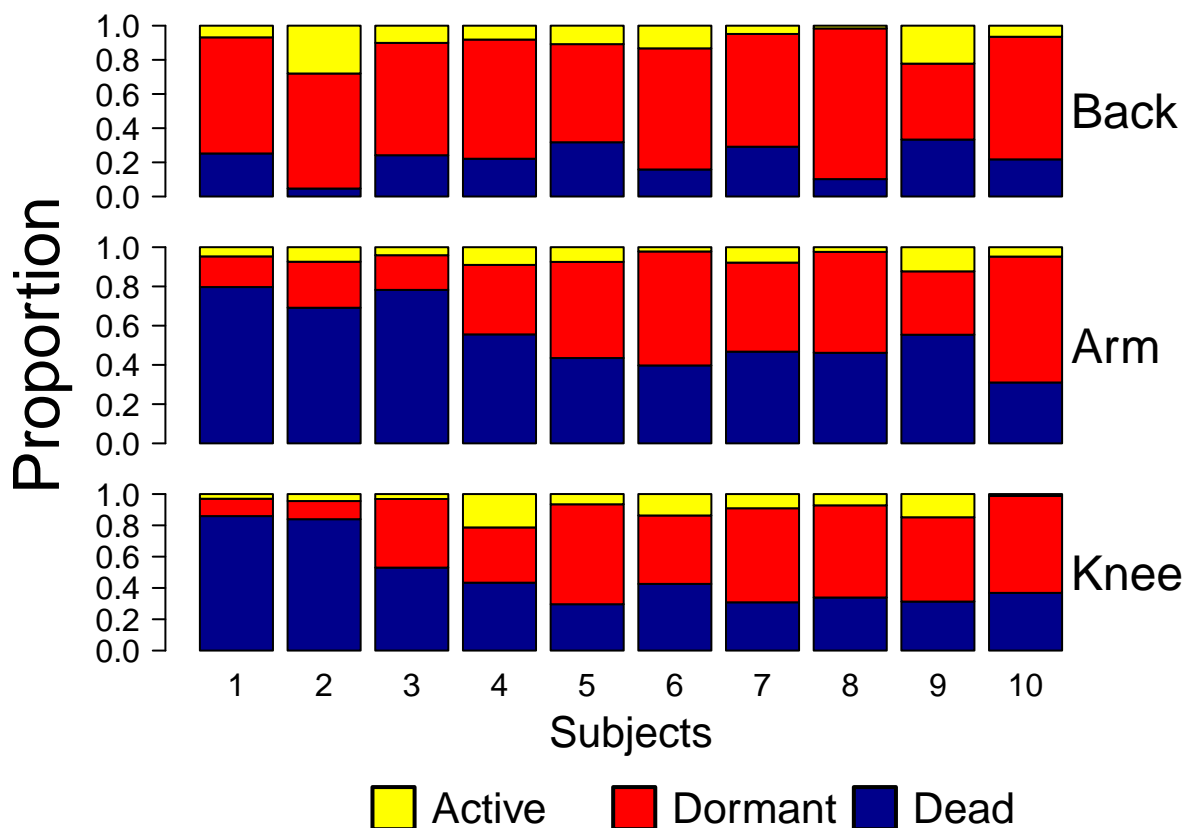
par(mar = c(1, 7, 1, 5.5))

barplot.arm <- barplot(as.matrix(arm.sub),
    ylim = c(0, 1), las = 1,
    beside = FALSE,
    cex.axis = 1.5,
    axisnames = FALSE,
    col = c("darkblue", "red", "yellow"), plot = TRUE)
mtext("Proportion", side = 2, line = 4, cex = 2)
mtext("Arm", side = 4, las = 1, cex = 1.5, adj = 0, line = -1)

par(mar = c(5, 7, 1, 5.5))
barplot.knee <- barplot(as.matrix(knee.sub),
    ylim = c(0, 1), las = 1,
    beside = FALSE,
    cex.axis = 1.5,
    xlab = "Subjects", cex.lab = 2, cex.names = 1.5,
    axisnames = TRUE,
    col = c("darkblue", "red", "yellow"), plot = TRUE)
mtext("Knee", side = 4, las = 1, cex = 1.5, adj = 0, line = -1)

par(mar=c(0, 9, 0, 0))
plot.new()
legend("center", c("Active","Dormant","Dead"), pt.lwd = 2, col = "black",
    pt.bg = c("yellow","red", "darkblue"), pch = c(22, 22, 22), bty='n',
    ncol = 3, cex = 2, pt.cex = 5)

```



E. SINGLE ANOVAs: do habitats vary in classes of activity?

```
# Make a factor table for habitats
hab.type <- hab.per$habitat

# ANOVA for active "are "
anova.act <- aov(act.per ~ hab.type)
summary(anova.act)
```

```
##           Df Sum Sq Mean Sq F value Pr(>F)
## hab.type    2  0.0126  0.00628    1.62   0.22
## Residuals  27  0.1049  0.00389
```

```
TukeyHSD(anova.act)
```

```
## Tukey multiple comparisons of means
## 95% family-wise confidence level
##
## Fit: aov(formula = act.per ~ hab.type)
##
## $hab.type
##           diff      lwr      upr p adj
## back-arm  0.05003 -0.01910 0.11915 0.1905
```

```
## knee-arm    0.02230 -0.04683 0.09142 0.7064
## knee-back -0.02773 -0.09686 0.04139 0.5864
```

```
anova.dorm <- aov(dorm.per ~ hab.type)
summary(anova.dorm)
```

```
##              Df Sum Sq Mean Sq F value    Pr(>F)
## hab.type      2  0.588   0.2942      11 0.00032 ***
## Residuals    27  0.723   0.0268
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
TukeyHSD(anova.dorm)
```

```
##    Tukey multiple comparisons of means
##      95% family-wise confidence level
##
## Fit: aov(formula = dorm.per ~ hab.type)
##
## $hab.type
##              diff          lwr          upr    p adj
## back-arm -0.32702 -0.50850 -0.1455 0.0004
## knee-arm -0.07383 -0.25530  0.1077 0.5779
## knee-back 0.25319  0.07171  0.4347 0.0050
```

```
anova.dead <- aov(dead.per ~ hab.type)
summary(anova.dead)
```

```
##              Df Sum Sq Mean Sq F value    Pr(>F)
## hab.type      2  0.434   0.2170      8.14 0.0017 **
## Residuals    27  0.719   0.0266
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
TukeyHSD(anova.dead)
```

```
##    Tukey multiple comparisons of means
##      95% family-wise confidence level
##
## Fit: aov(formula = dead.per ~ hab.type)
##
## $hab.type
##              diff          lwr          upr    p adj
## back-arm  0.27699  0.09599  0.45799 0.0021
## knee-arm  0.05153 -0.12947  0.23253 0.7621
## knee-back -0.22546 -0.40646 -0.04446 0.0124
```

### C. Test for differences among habitats using PERMANOVA

```

# Subset activity data
hab.per.cell <- hab.per[,3:5]
hab.type <- hab.per$habitat
subjects <- hab.per$subject

# Run PERMANOVA with adonis function blocking by subject
adonis(hab.per.cell ~ hab.type, method = "bray", permutations = 999, strata = subjects)

##
## Call:
## adonis(formula = hab.per.cell ~ hab.type, permutations = 999,          method = "bray", strata = subject
##
## Blocks: strata
## Permutation: free
## Number of permutations: 999
##
## Terms added sequentially (first to last)
##
##              Df SumsOfSqs MeanSqs F.Model    R2 Pr(>F)
## hab.type      2      0.579  0.2896   8.85 0.396  0.001 ***
## Residuals    27      0.884  0.0327         0.604
## Total        29      1.463          1.000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

### 3) ANALYSIS OF DIFFERENT AGES

#### A. Load data

```

# Load raw data
age.raw <- read.table("skin.age.txt", sep = "\t", header = TRUE)

# Remove potential outlier
age <- age.raw[ ! age.raw$Sample == "G1", ]

# Look at raw data structure
str(age)

## 'data.frame': 26 obs. of 6 variables:
## $ Sample : Factor w/ 27 levels "A1","A2","A4",...: 1 2 3 4 5 6 7 8 9 10 ...
## $ Age : int 3 2 7 11 20 24 22 25 23 27 ...
## $ RelAct : num 0.1263 0.0433 0.1058 0.0316 0.0159 ...
## $ RelDead : num 0.4056 0.3975 0.4471 0.0926 0.8822 ...
## $ RelDorm : num 0.468 0.559 0.447 0.876 0.102 ...
## $ TotalCells: int 673 1132 416 443 7650 1611 354 135 4302 296 ...

# Linear regression
fit <- lm(RelAct ~ Age, data = age)
summary(fit)

```

```
##
## Call:
## lm(formula = RelAct ~ Age, data = age)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.06399 -0.02728 -0.00179  0.01303  0.14787
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.102298   0.017911   5.71    7e-06 ***
## Age         -0.001118   0.000446  -2.51    0.019 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0437 on 24 degrees of freedom
## Multiple R-squared:  0.208, Adjusted R-squared:  0.175
## F-statistic:  6.3 on 1 and 24 DF, p-value: 0.0192
```

```
# Plot activity against age
plot(age$Age, age$RelAct, ylim = c(0, 0.25), xlim = c(-0.1, 75),
      xlab = expression(paste("Age")),
      ylab = "Proportion Active", las = 1,
      xaxt = "n", yaxt = "n")

axis(side=1, at = c(0, 10, 20, 30, 40, 50, 60, 70), las=1)
axis(side=2, at = c(0,0.1,0.2), las=1)

# Add regression line
newAge <- seq(min(age$Age), max(age$Age), 1)
regline <- predict(fit, newdata = data.frame(Age = newAge))
lines(newAge, regline)

# Add 95% confidence intervals
conf95 <- predict(fit, newdata = data.frame(Age = newAge),
                  interval = c("confidence"), level = 0.95, type = "response")
matlines(newAge, conf95[, c("lwr", "upr")], type="l", lty = 2, lwd = 1, col = "black")
```

