

```

#####
# Estimating moss mat density/biomass -- 18 Oct 2014
# Rob Smith, Oregon State University / USDA-USFS-FHM Program
# Contents:
# I. Data pre-processing
# II. Summary stats
# III. CN elemental analysis
# IV. Exponential decay model: given a depth, predict density
# V. Hectare-scale biomass/C/N estimates
# VI. Bootstrap resampling of biomass
# VII. Functional group richness, composition, dispersion
#####
#### I. Data pre-processing #####
rm(list=ls())
df <- read.csv("2013biomass.csv",header=T) # calibration set
df2 <- read.csv("2013sites.csv", header=T) # site descriptors
df3 <- read.csv("2013plots.csv", header=T) # implementation set
sem <- function(x) sd(x, na.rm=T)/sqrt(length(na.omit(x))-1) # standard error

# pre-process biomass data in df
vol <- df$depth * df$area
dens <- df$ovendrymass / vol
cn <- df$c / df$n
df <- cbind(df, vol, dens, cn)
rm(list=c("vol", "dens", "cn"))

# pre-process plot data in df3
df3$cover <- df3$volume <- df3$cvalue <- df3$nvalue <- NA
df3$cnvalue <- df3$shabitat <- df$shabitat <- NA
df3$shabitat <- df2$shabitat[match(df3$plotid, df2$plotid)] # assign habitat
df$shabitat <- df2$shabitat[match(df$plotid, df2$plotid)] # assign habitat
df3$type <- ifelse(df3$gf %in% c("A","P","V","F","S"), "moss", "lich")

# re-order factor levels for later plotting
df$shabitat <- factor(df$shabitat,levels=c("upland", "lowland", "mixed",
"alpine", "coastal", "montane", "dry forest", "steppe"))
df2$shabitat <- factor(df2$shabitat,levels=c("upland", "lowland", "mixed",
"alpine", "coastal", "montane", "dry forest", "steppe"))
df3$shabitat <- factor(df3$shabitat,levels=c("upland", "lowland", "mixed",
"alpine", "coastal", "montane", "dry forest", "steppe"))

# convert cover classes to midpoint percentage values
df3$cover[(df3$coverclass==0)] <- 0
df3$cover[(df3$coverclass==1)] <- 0.001
df3$cover[(df3$coverclass==2)] <- 0.005
df3$cover[(df3$coverclass==3)] <- 0.015
df3$cover[(df3$coverclass==4)] <- 0.035
df3$cover[(df3$coverclass==5)] <- 0.075
df3$cover[(df3$coverclass==6)] <- 0.175
df3$cover[(df3$coverclass==7)] <- 0.375

```

```

df3$cover[(df3$coverclass==8)] <- 0.625
df3$cover[(df3$coverclass==9)] <- 0.85
df3$cover[(df3$coverclass==10)] <- 0.975
df3$cover <- df3$cover * 1000      # convert plot cover to square cm
df3$volume <- df3$depth * df3$cover # compute plot volume in cubic cm

# make acrocarps less than 1cm tall == "crust" category
df$gf <- as.character(df$gf)
df$gf[ df$depth <= 1 & df$gf=="A" ] <- "C"
df$gf <- as.factor(df$gf)
# and again:
df3$gf <- as.character(df3$gf)
df3$gf[ df3$depth <= 1 & df3$gf=="A" ] <- "C"
df3$gf <- as.factor(df3$gf)

require(sciplot) # for plotting
require(ggplot2) # for plotting
require(ply)    # for data munging

require(reshape) # for data munging
require(gridExtra) # for arranging multiple ggplots
require(scales)  # for log-log scales
require(pastecs) # for descriptive statistics
require(vegan)   # for NMS and other ecological tasks
require(car)     # for Type III ANOVA

# set theme attributes for plotting
mytheme <- theme_bw() + theme(legend.position='none',
  panel.grid.minor=element_blank(),
  panel.grid.major=element_blank(),
  panel.background=element_blank(),
  axis.text.x =element_text(size=12, hjust=0),
  axis.text.y =element_text(size=12, angle=0),
  axis.title.x=element_text(size=18),
  axis.title.y=element_text(size=18, angle=90),
  strip.text.x=element_text(size=18) )
#####
#### II. Summary stats #####

# fxnl richness *per site* (sampling effort uniform among 33 microplots)
dftemp <- df3[df3$cover!=0, ]      # microplots with non-zero cover
count3 <- as.matrix(table(dftemp$plotid, dftemp$microplot))
fxnlrich.m <- apply(count3, 1, FUN=mean) # *avg* microplot richness per plot
spe <- read.csv("nms_spe.csv", header=T, colClasses="numeric") # wide format spp
spe[spe >0 ] <- 1
fxnlrich <- as.vector(apply(spe, 1, sum)) # *sum* plot richness
fxnlH <- diversity(spe, index = "shannon")# Shannon entropy using base of ln
plotsum <- as.data.frame(cbind(fxnlrich.m, fxnlrich, fxnlH))
plotsum$rownames <- rownames(plotsum)
# *avg* fxnlrich per microplot
df2$fxnlrich.micro <- plotsum$fxnlrich.m[match(df2$plotid, plotsum$rownames)]
# *sum* fxnlrich per site
df2$fxnlrich <- plotsum$fxnlrich [match(df2$plotid, plotsum$rownames)]
# # fxnlH per site

```

```

df2$fxnlH <- plotsum$fxnlH [match(df2$plotid, plotsum$rownames)]
rm(spe, dftmp, plotsum, count3, fxnlrich.m, fxnlrich, fxnlH)
# mean sampling times by habitat (sampling effort not uniform)
options(scipen=10, digits=3)
aggregate(df2$time, list(Habitat=df2$habitat), FUN=function(x)stat.desc(x))
stat.desc(df2$time) # pooled
# mean fxnl rich by habitat
aggregate(df2$fxnlrich, list(Habitat=df2$habitat), FUN=function(x)stat.desc(x))
stat.desc(df2$fxnlrich) # pooled
# mean cover by habitat
aggregate(df3$cover, list(Habitat=df3$habitat), FUN=function(x)stat.desc(x))
stat.desc(df3$cover) # pooled
# mean depth by habitat
aggregate(df3$depth, list(Habitat=df3$habitat), FUN=function(x)stat.desc(x))
stat.desc(df3$depth) # pooled

# do sample times differ among habitats? Unbalanced replication, need Type-3 SS.
Anova(aov(df2$time ~ df2$habitat), type=3)
Anova(aov(df2$fxnlrich ~ df2$habitat), type=3)
#####
#### III. CN elemental analysis: percentages by functional group #####
# plot fxnl grp means
w <- data.frame(value1 = df$n, value2 = df$c, value3=df$cn,
                factor1 = as.factor(df$gf), factor2=as.factor(df$year))
w <- ddpby(w, .(factor1), function(d)
  data.frame(v1mean = mean(d$value1, na.rm=T), v1sem = sem(d$value1),
            v2mean = mean(d$value2, na.rm=T), v2sem = sem(d$value2),
            v3mean = mean(d$value3, na.rm=T), v3sem = sem(d$value3)))
(p1 <- ggplot() + mytheme + xlim(0.4,3) + ylim(38.5,47.5) +
  geom_point(data = w, aes(x = v1mean, y = v2mean), size=1.5) +
  geom_errorbar (data = w, width=0.07,
                aes(x=v1mean, ymin=v2mean-v2sem, ymax=v2mean+v2sem) ) +
  geom_errorbarh(data = w,aes(x=v1mean, y=v2mean, xmin=v1mean-v1sem,
                             xmax=v1mean+v1sem), height = 0.25) +
  annotate("text", x=w$v1mean-0.07, y=w$v2mean+0.5,
          label=paste(LETTERS[c(5,7,8,4,2,6,1,3)]), size=4) +
  xlab("Total nitrogen (%)")+ ylab("Organic carbon (%)") )
rm(w, p1)

# does c/n/cn/biomass (in the calibration set) differ by fxnl group?
dftmp <- df[which(df$c!="NA"),] # drop microplots that lack C% estimates
Anova(aov(dftmp$c ~ dftmp$gf), type=3) # no
Anova(aov(dftmp$n ~ dftmp$gf), type=3) # yes
Anova(aov(dftmp$cn ~ dftmp$gf), type=3) # yes
Anova(aov(dftmp$ovendrymass ~ dftmp$gf), type=3) # not for most pairwise grps
# therefore, must assign N and C/N by fxnl group means:
# mean n% by fxnl grp
(meann <- data.frame(n=sapply(tapply(df$n, df$gf, stat.desc), `[`,9)))
# mean cn by fxnl grp
(meancn <- data.frame(cn=sapply(tapply(df$cn, df$gf, stat.desc), `[`,9)))
dftmp <- cbind(meann, meancn)
df3$value <- mean(dftmp$c) # assign c%
df3$nvalue <- dftmp$n [match(df3$gf, rownames(dftmp))] # assign n%
df3$cnvalue <- dftmp$cn[match(df3$gf, rownames(dftmp))] # assign cn ratio

```

```

df3[df3$gf %in% c("0","V"), colnames(df3)== "cvalue"] <- 0 # replace NA with 0
df3[df3$gf %in% c("0","V"), colnames(df3)== "nvalue"] <- 0 # replace NA with 0
df3[df3$gf %in% c("0","V"), colnames(df3)== "cnvalue"]<- 0 # replace NA with 0
rm(dftemp, dftmp, meann, meancn)
#####
#### IV. Exponential decay model: given a depth, predict density #####
# first, model selection using NPMR software HyperNiche (McCune and Mefford)
f1 <- function(x,Const,a,b) {Const + a*exp(b * x)} # model function
f2 <- deriv3( ~ Const + a * exp(-b * depth), # model function
             c("Const","a","b"), function(Const, a, b, depth) NULL)
summary(m1 <- nls(dens ~ f1(depth, Const, a, b), # model fit
                data=df, start=list(Const=0.5,a=0.01,b=-0.5)))
coefs <- coef(m1)
summary(m2 <- nls(dens~f2(Const,a,b,depth), data=df,
                start=list(Const=0.5,a=0.01,b=0.5)))
Height <-seq(0, 30, by=0.1)
se.fit <- sqrt(apply(attr(predict(m2,list(depth = Height)),"gradient"),1,
                    function(Height) sum(vcov(m2)*outer(Height,Height))))
predse <- predict(m1,list(depth=Height))+outer(se.fit, qnorm(c(.5,.025,.975)))
confint <- data.frame(x=Height,y=predse[,1],ymin=predse[,2],ymax=predse[,3])

# plotting
p1 <- ggplot(data=df,aes(x=depth, y=dens))+
  mytheme + theme(panel.border=element_blank(), axis.line = element_line()+
  geom_point(colour='black')+
  stat_function(fun=f1,args=list(
    Const=coefs[1],a=coefs[2],b=coefs[3]),size=1)+
  geom_ribbon(data=confint, aes(x=x,y=y,ymax=ymax,ymin=ymin),alpha=0.2)+
  scale_x_continuous("Depth (cm)")+
  scale_y_continuous(expression(paste("Bulk density (g cm" ^-3,")",sep=")))
p2 <- ggplot(data=df,aes(x=vol, y=ovendrymass))+
  mytheme + geom_point(colour='black', size=1)+
  scale_x_log10(limits=c(7,11500),
               expression(paste("Volume (cm" ^3,")", sep=")),
               breaks = trans_breaks("log10", function(x) 10^x),
               labels = trans_format("log10", math_format(10^.x))) +
  scale_y_log10(limits=c(0.4,700), expression(paste("Mass (g)")),
               breaks = trans_breaks("log10", function(x) 10^x),
               labels = trans_format("log10", math_format(10^.x))) +
  annotation_logticks()
vp <- viewport(width = 0.5, height = 0.5, x = 0.75, y = 0.75) # inset 2 plots
print(p1) ; print(p2, vp = vp)
rm(Height, p1, p2, vp, confint, coefs, predse, se.fit)
#####
#### V. Hectare-scale biomass/C/N estimates #####
# use neg exp model to predict density for implementation plots
df3[df3$coverclass=="0", colnames(df3)== "type"] <- "none" # replace NA w "none"
df3$type <- as.factor(df3$type)
newx <- list(depth=df3$depth)
df3$pred.dens <- df3$pred.mass <- df3$predC <- df3$predN <- NA
df3$pred.dens <- predict(m2, newdata=newx) # predicted density (g/cm^3)
df3$pred.dens <- df3$pred.dens[1:length(df3$pred.dens)] # get values only
rm(m1, m2, newx, f1, f2)
# compute microplot biomass/C/N as a function of density

```

```

df3$pred.mass <- df3$pred.dens * df3$volume # mass (g/microplot)
df3$predC <- df3$pred.mass * df3$cvalue/100 # total C (g/microplot)
df3$predN <- df3$pred.mass * df3$nvalue/100 # total N (g/microplot)

# convert microplot estimates to *hectare-level* estimates (within-sites)
dfz <- aggregate(df3$pred.mass, list(Plot=df3$plotid, Type=df3$type), mean)
xtable<- as.matrix( xtabs(dfz$x ~ ., data=dfz) )
xsums <- data.frame(margin.table(xtable, 1))
df8 <- as.data.frame(cbind(xtable, xsums$Freq))
colnames(df8) <- c("lichen","moss","none","combo")
df8 <- df8 * 100000 * 0.001 # there are 100000 Daub per ha, and 1000 g per kg
df8$plotid <- rownames(df8)
df8$habitat <- df2$habitat[match(df8$plotid, df2$plotid)] # assign habitat
dfz <- aggregate(df3$predC, list(Plot=df3$plotid), mean) # for *C*
kg_C <- as.vector(dfz$x) * 100000 * 0.001
kg_C <- kg_C[ -which(is.na(kg_C)) ]
dfz <- aggregate(df3$predN, list(Plot=df3$plotid), mean) # for *N*
kg_N <- as.vector(dfz$x) * 100000 * 0.001
kg_N <- kg_N[ -which(is.na(kg_N)) ]
df8 <- as.data.frame( cbind(df8, kg_C=kg_C, kg_N=kg_N ))
# write results to df2
df2$moss_kg <- df8$moss[match(df2$plotid, df8$plotid)] # assign by plot
df2$lich_kg <- df8$lich[match(df2$plotid, df8$plotid)] # assign by plot
df2$all_kg <- df8$combo[match(df2$plotid, df8$plotid)]# assign by plot
df2$kg_C <- df8$kg_C[match(df2$plotid, df8$plotid)] # assign by plot
df2$kg_N <- df8$kg_N[match(df2$plotid, df8$plotid)] # assign by plot
str(df2)
# summary stats
sumstats <- data.frame(
  Habitat= levels(df2$habitat),
  N = aggregate(df2$all_kg, list(habitat=df2$habitat), length)$x ,
  kg_ha = aggregate(df2$all_kg, list(habitat=df2$habitat), mean,na.rm=T)$x ,
  SE_ha = aggregate(df2$all_kg, list(habitat=df2$habitat), sem)$x ,
  kg_moss=data.frame(aggregate(df2$moss_kg,list(habitat=df2$habitat),
    mean,na.rm=T))$x,
  SE_moss= aggregate(df2$moss_kg, list(habitat=df2$habitat), sem)$x ,
  kg_lich=data.frame(aggregate(df2$lich_kg,list(habitat=df2$habitat),
    mean,na.rm=T))$x,
  SE_lich= aggregate(df2$lich_kg, list(habitat=df2$habitat), sem)$x ,
  kg_C = aggregate(df2$kg_C, list(habitat=df2$habitat), mean,na.rm=T)$x ,
  SE_C = aggregate(df2$kg_C, list(habitat=df2$habitat), sem )$x ,
  kg_N = aggregate(df2$kg_N, list(habitat=df2$habitat), mean,na.rm=T)$x ,
  SE_N = aggregate(df2$kg_N, list(habitat=df2$habitat), sem )$x ,
  fxnlrich =aggregate(df2$fxnlrich, list(habitat=df2$habitat),
    mean,na.rm=T )$x,
  SE_fxnrich=aggregate(df2$fxnlrich, list(habitat=df2$habitat), sem )$x,
  fxnlH =aggregate(df2$fxnlH, list(habitat=df2$habitat), mean,na.rm=T)$x
)
(sumstats <- cbind(sumstats$Habitat, ### final table...
  (round(sumstats[,c(2:12)],0)),
  (round(sumstats[,c(13:14)],1)),
  (round(sumstats[15],3)) ))
rm(dfz, xtable, xsums, sumstats)

```

```

#####
#### VI. Bootstrap resampling of biomass #####
# fxn for bootstrapping -- return summary stats of replicated resamplings
bootmeans <- function(x, size, n){
  #create replicate data
  repdata <- as.data.frame(
    replicate(n=n, sample(x=x, size=size, replace=T)))
  # summary stats for each replicate
  means <- t(as.matrix(rowMeans(sapply(repdata, stat.desc))))
  dimnames(means)[[1]] <- paste("n",size,sep="")
  return(means)
}
# fxn to increment sample size
bootinc <- function(x, size, n){
  out <- matrix(nrow=33,ncol=14)
  repeat{
    out[size,] <- bootmeans( x=x, size=size, n=n )
    size <- size-1
    if(size==1){
      break
    }
  }
  colnames(out) <- c("nbr.val","nbr.null","nbr.na","min","max","range",
    "sum","median","mean","sem","CI.mean.0.95","var","sd","cv")
  out <- as.data.frame(out)
  return(out)
}
options(scipen=10)

# df4 is within-microplot biomass for all fxnl grps (later can scale to hectare)
df4 <- aggregate(pred.mass ~ microplot+habitat+plotid, data=df3, FUN=sum)
df4$transmass <- log10(df4$pred.mass+1) # log10(x+1) transform

### TIME WARNING ! ### runs 18 min when nreps=999
# nreps <- as.numeric(999) # number of replicates
# size <- as.numeric(33) # starting sample size
# system.time(
# r1 <- tapply(df4$pred.mass, df4$plotid, bootinc, size=size, n=nreps) #TIMEWARN!
# )
# r2 <- do.call("rbind", lapply(r1, data.frame, stringsAsFactors=F))
# r2 <- matrix(unlist(r2), ncol=ncol(r2), byrow=F, dimnames=dimnames(r2))
# r2 <- data.frame(r2)
# r3 <- cbind(df4, r2)
# r3$newcv <- r3$sem/r3$mean*100 # relative standard err (RSE)
# write.csv(r3, "result_999boots_biomass.csv") # write to file...
r3 <- read.csv("result_999boots_biomass_21Aug2014.csv",header=T) # or read from file
r3$habitat <- factor(r3$habitat, levels=
  c("upland", "lowland", "mixed", "alpine",
    "coastal", "montane", "dry forest", "steppe"))
# plot bootstrap results
mytran_trans <- function(x) trans_new(name="mytran",
  transform = function(x) log10(x+1),
  inverse = function(x) 10^(x)-1,

```

```

breaks = log_breaks(n=6,base=10) )
pA <- ggplot(data=r3, aes(x=microplot,y=mean))+
  mytheme + geom_point(aes(colour="black"), size=1.5) +
  geom_line(aes(colour="black"))+scale_colour_manual(values=c("black"))+
  geom_errorbar(width=0.5,colour="grey23",
    aes(ymin=mean-sem,ymax=mean+sem))+
  facet_wrap(~habitat, nrow=1)+ coord_trans(y="mytran")+
  scale_x_continuous("", labels=NULL) +
  scale_y_continuous(expression(paste("Mean mass (g 1000 cm-2,"),sep="")),
    trans = "mytran", limits=c(0.1,1000)) +
  theme(plot.margin=unit(c(0.1,0.2,-0.1,0.2), "in"))
pB <- ggplot(data=r3, aes(x=microplot,y=newcv))+
  mytheme + geom_point(aes(colour="black"), size=1.5) +
  geom_line(aes(colour="black")) + facet_wrap(~habitat, nrow=1)+
  scale_colour_manual(values=c("black")) +
  geom_abline(slope=0, intercept=25, linetype=2, colour="red") +
  scale_x_continuous(expression(paste(italic("N"), " microplots")))+
  scale_y_continuous(expression(paste(italic("RSE")," (% of mass",sep="")),
    limits=c(0,80) )+
  theme(plot.margin=unit(c(-0.1,0.2,0.1,0.2), "in"))
gA <- ggplotGrob(pA) ; gB <- ggplotGrob(pB)
maxwid <- grid::unit.pmax(gA$widths[2:5], gB$widths[2:5])
gA$widths[2:5] <- as.list(maxwid) ; gB$widths[2:5] <- as.list(maxwid)
windows(20,7)
grid.arrange(gA, gB, ncol=1)
rm(df4, r1, r2, r3, gA, gB, pA, pB, nreps, size, maxwid, mytran_trans)
dev.off()
#####
#### VII. Functional group richness, composition, dispersion ####
# In functional group space, not species space: NMS, PERMANOVA
# NOTE: final NMS solution used robust solutions from PC-ORD software (not R)
# see PC-ORD also for Monte Carlo sampling test for NMS
spe <- read.csv("nms_spe.csv", header=T) # requires data in wide format
id <- read.csv("nms_id.csv", header=T) # requires data in wide format
genlogtrans <- function(x) { # generalized log-transform function,
  wasDataFrame <- is.data.frame(x) # see McCune and Grace (2002) pg. 69
  x <- as.matrix(x)
  mins <- min(x[which(x>0)])
  const <- as.integer(log10(mins))
  d <- 10^const
  x <- log10(x + d) - const
  if (wasDataFrame)
    x <- as.data.frame(x)
  x
}
spe <- genlogtrans(spe) # transformed data
id$habitat <- df2$habitat[match(id$plotid, df2$plotid)] # assign habitat
id$habitat <- factor(id$habitat,ordered=T, levels=c( #re-order for plotting
  "coastal", "montane", "dry forest", "steppe",
  "alpine", "mixed", "upland", "lowland"))
d <- vegdist(spe, method = "bray", binary = F) # make distance matrix
(ord <- metaMDS(spe, dist="bray", k=2, trymax=99, autotransform=F,
  noshare=F, plot=F, wascores=T))
stressplot(ord, main="Stress Plot", cex.lab = 1.5)

```

```

# plot by habitat
scores <- data.frame(scores(ord, display = "sites"), habitat=id$habitat)
newcols <-c("blue", "lightblue", "green", "darkgreen", "red",
            "orange", "purple1", "yellow")
rownames(ord$species)<-c("TurfMoss", "CrustLich", "N-fixLich", "N-fixMoss",
                        "Lichen", "FeatherMoss", "Forage ", "Sphagnum", "Liverwort")
(p1 <- ggplot(data=scores,aes(x=NMDS1, y=NMDS2, group=scores$habitat))+
  mytheme + annotate("text", x=0.5, y=-1.15, label="Stress=13.2")+
  geom_point(aes(shape=scores$habitat,colour=scores$habitat),size=4)+
  guides(colour=guide_legend(override.aes = list(size=4)))+
  scale_shape_manual(values=c(17,17,17,17,19,19,19,19))+
  scale_colour_manual(values=c(newcols))+
  annotate("text", x=ord$species[,1], y=ord$species[,2],
          label=rownames(ord$species), size=6 )

# do habitats differ in trait composition?
(mod1 <- mrpp (d,id$habitat, perm = 9999))      # yes...
(mod2 <- adonis(d~id$habitat, perm = 9999))    # and yes

# do habitats differ in beta-trait diversity (trait turnover, dispersion)?
beta1 <- betadisper(d, id$habitat, type = "centroid")
permutest(beta1) # yes
#### end document #####

```