The accuracy and precisions of body mass estimation in non-avian dinosaurs

Nicolás E. Campione and David C. Evans

code compiled by: Nicolás E. Campione ([ncampion@une.edu.au](mailto:ncampion@une.edu.au))

The following code accompanies this paper and includes all analyses. Note that the code should run as is, by copy-pasting the code boxed into R, and assuming all files are in the same working directory, although file names may need to be adjusted. Hash (#) can be toggled off to run specific lines.

# Set-up

## Required libraries

#install.packages(c("readxl", "MASSTIMATE", "RColorBrewer", "smatr", "car", "geomorph"))  
library(readxl)  
library(MASSTIMATE)  
library(RColorBrewer)  
library(smatr)  
library(car)  
library(geomorph)

## Body Size Database

For information about the database, including heading and worksheet explanations, please see the associated ReadMe file.

#Volumetric and Measurement data set  
bs.data <- read\_xlsx("Campione&Evans\_Appendix\_S1.xlsx",sheet = 1)  
bs.data <- bs.data[-which(bs.data$Method == "MCH" & bs.data$BM.Qual == "Minimum"),] #removes minimum MCH estimates  
bs.data <- droplevels(bs.data[which(bs.data$Match == "?" | bs.data$Match == "x"),]) #final dataset  
#bs.data <- droplevels(bs.data[which(bs.data$Match == "x"),]) #perfect match dataset, will over-write previous line, 262 entries  
tapply(bs.data$Clade.1, bs.data$Clade.1, length) # counts for major clades

## Ornithischia Sauropodomorpha Theropoda   
## 123 120 189

## Database Exploration

This section of code relates to Fig. 5, which explores the various aspects of the database that will be subsequently tested using our proposed comparative framework. This code will save the plot to a pdf file in the working directory.

#set-up  
pdf("Figure 5.pdf", width = 7.0866142, height = 9)  
split.screen(c(3,1)) #screens 1 to 3

## [1] 1 2 3

split.screen(c(1,2), screen = 2) #screen 2 split into screens 4 and 5

## [1] 4 5

split.screen(c(1,2), screen = 3) #screen 3 split into 6 and 7

## [1] 6 7

#by clade  
clade.1.col <- c("#006e8d", "#ad1457", "#263802")  
clade.1.counts <- tapply(as.factor(bs.data$Clade.1), INDEX = as.factor(bs.data$Clade.1), length)  
#par(mar = c(7,4,1,1))  
#barplot(clade.1.counts, col = clade.1.col, las = 3, border = NA, cex.names = 0.75)  
clade.2.col <- c("#006e8d", "#2b99b8", "#50b1cc", "#69c2d8", "#96d4e1",   
 "#ad1457", "#c62828", "#d81b60", "#f44336", "#f06292", "#ef9a9a", "#f8bbd0",  
 "#263802", "#0c6600", "#5f9207", "#0cbf00", "#7ad212", "#22ef3a", "#88f231", "#92ffba", "#c0ff93")  
clade.2.order <- c("Ornithischia", "Ankylosauria", "Marginocephalia", "Ornithopoda", "Stegosauria",  
 "Sauropodomorpha", "Brachiosauridae", "Diplodocoidea", "Eusauropoda", "Macronaria", "Neosauropoda", "Titanosauroidea",   
 "Theropoda", "Allosauroidea", "Carcharodontosauridae", "Ceratosauria", "Coelophysoidea", "Coelurosauria", "Maniraptoriformes", "Megalosauroidea", "Tyrannosauroidea")  
names(clade.2.col) <- clade.2.order  
clade.2.counts <- tapply(as.factor(bs.data$Clade.2), INDEX = as.factor(bs.data$Clade.2), length)  
screen(1)  
par(mar = c(8,4,1,1))  
clade.2.counts[c(15, 17, 19)] <- clade.1.counts #use total counts for main clades  
barplot(clade.2.counts[clade.2.order], col = clade.2.col, las = 3, border = NA, cex.names = 0.75)  
  
#by method  
screen(4)  
par(mar = c(8,4,1,1))  
method.counts <- tapply(as.factor(bs.data$Method), INDEX = as.factor(bs.data$Method), length)  
method.order <- c("Physical", "GDI", "Polynomial", "Slicing", "MCH", "Photogrammetry", "Scan")  
method.col <- brewer.pal(length(method.order), "Dark2"); names(method.col) <- method.order  
barplot(method.counts[method.order], col = method.col, las = 3, border = NA, cex.names = 0.75)  
  
#by year  
year.counts <- tapply(as.factor(bs.data$EBM.yr), INDEX = as.factor(bs.data$EBM.yr), length)  
year.mat <- xtabs(~Method+EBM.yr, data = bs.data)  
screen(6)  
par(mar = c(4,4,1,1))  
barplot(year.mat[method.order,], las = 3, border = NA, cex.names = 0.75, col = brewer.pal(length(method.order), "Dark2"))  
  
#by Author(s)  
author.counts <- tapply(as.factor(bs.data$EBM.ref), INDEX = as.factor(bs.data$EBM.ref), length)  
author.meths <- tapply(bs.data$Method, INDEX = as.factor(bs.data$EBM.ref), function(x) return(x[1]))  
screen(5)  
par(mar = c(6,4,1,1))  
barplot(author.counts, col = method.col[author.meths],las = 3, border = NA, cex.names = 0.5)  
  
#by perfect match  
partial.match <- length(which(bs.data$Match == "?"))  
perfect.match <- length(which(bs.data$Match == "x"))  
screen(7)  
par(mar = c(1,1,1,1))  
pie(x = c(partial.match, perfect.match), labels = c("Partial Match", "Perfect Match"), border = "white", col = c("seagreen2", "deepskyblue"))  
close.screen(all.screens = TRUE)  
dev.off()

## png   
## 2

## Limb Data Sets

Extant data are sourced from Campione & Evans (2012) and dinosaur data are from Benson *et al.* (2018). These data are also used to generate Fig. 4, which is saved as a PDF in the working directory.

# download extant limb data from BMC Biology  
download.file("https://static-content.springer.com/esm/art%3A10.1186%2F1741-7007-10-60/MediaObjects/12915\_2012\_575\_MOESM1\_ESM.XLS", destfile = "CE\_data.xls", quiet = TRUE, mode = "wb")  
extant.data <- read\_xls("CE\_data.xls", sheet = 1, range = cell\_rows(1:256)) #download data from Campione & Evans 2012  
extant.data <- extant.data[extant.data$Family != "Talpidae" & extant.data$`Higher Clade` != "Amphibia",] #remove amphibian and mole components of data set  
  
# download dinosaur limb data from Palaeontology  
download.file("https://datadryad.org/stash/downloads/file\_stream/75310", "Benson\_data.zip", mode = "wb")  
unzip("Benson\_data.zip")  
benson.data <- read\_xls("Dataset S1/Dataset S1.xls", sheet = 1, na = "NA")  
# Add gait column for sauropodomorphs and ornithischians  
benson.data <- data.frame(benson.data, Gait = c(NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, "quad", "quad", "quad", NA, NA, "quad", "quad", NA, NA, NA, "quad", NA, "quad", "quad", "quad", "quad", NA, NA, "quad", NA, NA, "quad", NA, NA, "quad", NA, NA, NA, NA, NA, "quad", NA, NA, NA, NA, "quad", NA, NA, NA, "quad", "quad", NA, "quad", "quad", NA, "bip", NA, NA, "quad", "quad", "quad", NA, "bip", NA, NA, "quad", NA, NA, "quad", NA, "quad", NA, NA, NA, "quad", "quad", NA, "quad", "quad", "quad", NA, "quad", NA, NA, NA, "bip", "quad", "quad", "quad", NA, "bip", "quad", "quad", NA, NA, NA, "quad", NA, "quad", NA, "quad", NA, NA, "quad", "quad", NA, "quad", NA, NA, NA, "quad", NA, "quad", NA, "quad", NA, NA, NA, "quad", NA, NA, NA, NA, NA, NA, NA, "quad", "quad", NA, NA, "quad", NA, NA, NA, "quad", "bip", NA, "quad", NA, NA, "quad", "quad", "bip", NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, "quad", NA, NA, "quad", "quad", NA, "quad", "bip", NA, NA, NA, NA, "quad", NA, NA, NA, NA, NA, NA, "bip", NA, NA, NA, NA, NA, NA, NA, "quad", "quad", NA, "quad", "quad", "bip", NA, "bip", "quad", "quad", "quad", NA, "bip", NA, NA, NA, NA, NA, "quad", NA, NA, NA, "quad", NA, "quad", NA, "quad", NA, NA, NA, "quad", "quad", NA, "quad", NA, NA, NA, "quad", NA, NA, NA, "quad", NA, "bip", "quad", rep(NA, 447), "quad", NA, "quad", NA, NA, "quad", NA, NA, NA, NA, NA, NA, NA, NA, NA, "bip", NA, NA, NA, "quad", "quad", NA, NA, NA, NA, NA, NA, NA, NA, NA, "quad", NA, "quad", NA, NA, NA, NA, NA, "quad", "quad", "quad", NA, NA, "quad", NA, NA, NA, NA, NA, "quad", "quad", NA, NA, NA, NA, NA, "quad", NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, "bip", "bip", "bip", NA, "quad", NA, "quad", "quad", NA, "bip", NA, NA, NA, NA, NA, "quad", NA, "bip", NA, "quad", "bip", "quad", NA, "quad", "quad", NA, NA, "quad", NA, NA, NA, NA, "quad", NA, "quad", NA, NA, NA, NA, NA, NA, NA, "bip", NA, "quad", NA, NA, "quad", NA, "quad", "quad", NA, "bip", NA, "quad", "quad", NA, "bip", NA, NA, NA, "quad", "quad", NA, NA, NA, NA, NA, NA, NA, "quad", NA, NA, "quad", "bip", NA, NA, "quad", "quad", "quad", NA, NA, NA, NA, "quad", NA, "quad", NA, "bip", NA, NA, NA, NA, "bip", NA, "quad", "quad", NA, NA, NA, "quad", NA, NA, NA, "quad", NA, NA, NA, "quad", "bip", NA, NA, "bip", NA, "bip", "quad", NA, NA, NA, NA, NA, "quad", "quad", "bip", NA, NA, NA, NA, "quad", NA, NA, NA, NA, "quad", NA, "bip", NA, NA, NA, NA, "quad", "quad", NA, NA, NA, "bip", NA, "bip", NA, NA, NA, NA, NA, "quad", NA, NA, "quad", "quad", "bip", NA, "quad", NA, NA, "quad", NA, NA, NA, NA, NA, NA, NA, NA, "bip", NA, "quad", "quad", "quad", NA, NA, "quad", "quad", NA, NA, "quad", NA, NA, NA, NA, NA, "quad", NA, NA, NA, NA, "bip", "bip", NA, NA, NA, "quad", "quad", "quad", "quad", NA, "quad", NA, NA, NA, NA, NA, NA, "quad", NA, "quad", NA, "bip", NA, NA, NA, NA, NA, NA, NA, "bip", NA, "bip", "bip", NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, "quad"))  
  
# HC vs FC comparisons  
names(extant.data)[c(1, 6, 8, 10)] <- c("Mon.Groups", "BM", "HC", "FC") #modify column names for ease of analysis  
extant.data <- data.frame(extant.data, HCFC = extant.data$HC+extant.data$FC, log.BM = log10(extant.data$BM), log.HCFC = log10(extant.data$HC+extant.data$FC))  
circ.sma <- sma(HC~FC\*Mon.Groups, data = extant.data, log = "xy")  
sma.col <- c("blue2", "goldenrod1", "firebrick2", "chartreuse4",   
 "deepskyblue1", "darkorange1", "darkorchid3", "darkgrey",   
 "mediumpurple1", "orangered2", "chocolate", "burlywood3",   
 "goldenrod4", "darkolivegreen2", "palevioletred3",   
 "darkseagreen3", "sandybrown", "tan", "gold", "violetred4",   
 "darkgreen")  
pdf("Figure 4.pdf", width = 7.0866142, height = 6) #activate to produce pdf of figure 3  
par(mar = c(5,4,1,1))  
plot(circ.sma, pch = 19, xlab = "Femur circumference (mm, log scale)", ylab = "Humerus circumference (mm, log scale)", xlim = c(5,1000), ylim = c(4,800))  
legend("topleft",legend = circ.sma$groups, col = sma.col[1:9], pch = 19)  
dino.sma <- sma(HC~FC\*as.factor(Clade), data = benson.data[benson.data$Clade != "Theropoda",], log = "xy")  
plot(dino.sma, pch = 15, col = sma.col[20:21], add = TRUE)  
legend("bottomright", legend = dino.sma$groups, col = sma.col[20:21], pch = 15)  
dev.off()

## png   
## 2

## ES Body Mass Estimates

ES body masses are estimated using either the appropriate bipedal or quadrupedal equations, the quadratic correction, and, in the case of the quadrupeds, the multiple regression equation.

dinos.mass <- matrix(nrow = nrow(bs.data),ncol = 12)  
colnames(dinos.mass) <- c('logQE','QE','QE-25%','QE+25%','logqQE','qQE','qQE-25%','qQE+25%','logMR','MR','MR-25%','MR+25%')  
for(i in 1:nrow(bs.data)) {  
 if(bs.data$Gait[i] == "quad") {  
 dinos.mass[i,1:8] <- QE(HFC = bs.data$HCFC[i], equation = "raw", quadratic = TRUE)  
 dinos.mass[i,9:12] <- MR(bs.data$HC[i], bs.data$FC[i], equation = "raw")  
 }  
 if(bs.data$Gait[i] == "bip") {  
 dinos.mass[i,1:8] <- cQE(FC = bs.data$FC[i],equation = "raw", quadratic = TRUE)  
 }  
}  
bs.data <- cbind(bs.data,dinos.mass)

Measures of deviation, ppe = per cent prediction error and res = residuals.

dinos.ppe <- ppe(true = bs.data$BM.g,pred = bs.data$QE)  
dinos.res <- log10(bs.data$BM.g)-bs.data$logQE

### QE vs. MR Estimates

In order to compare the estimates derived from the multiple regression (MR) and the preferred quadrupedal equation (QE), we tested for a statistical difference in the rank of each estimate (i.e. non-parametric test) using a Kruskal-Wallis Rank Sum Test (KW test). No significant difference was recovered in either ornithischians or sauropodomorphs.

#ornithischians  
orn.taxa <- levels(as.factor(bs.data$Taxon[bs.data$Clade.1 == "Ornithischia"& bs.data$Gait == "quad"]))  
orn.estimates <- matrix(nrow = length(orn.taxa), ncol = 2); rownames(orn.estimates) <- orn.taxa; colnames(orn.estimates) <- c("MR", "QE")  
for(i in orn.taxa) {  
 taxon <- which(bs.data$Taxon == i)  
 if(length(taxon) > 1) row <- taxon[1]  
 else row <- taxon  
 orn.estimates[i, "MR"] <- bs.data[row, "MR"]  
 orn.estimates[i, "QE"] <- bs.data[row, "QE"]  
}  
mean(orn.estimates[,1]-orn.estimates[,2], na.rm = TRUE)

## [1] -921838

kruskal.test(mass~model, data = data.frame(mass = c(as.vector(orn.estimates)), model = as.factor(c(rep("MR", length(orn.taxa)), rep("QE", length(orn.taxa))))))

##   
## Kruskal-Wallis rank sum test  
##   
## data: mass by model  
## Kruskal-Wallis chi-squared = 2.0423, df = 1, p-value = 0.153

#sauropodomorphs  
sau.taxa <- levels(as.factor(bs.data$Taxon[bs.data$Clade.1 == "Sauropodomorpha"& bs.data$Gait == "quad"]))  
sau.estimates <- matrix(nrow = length(sau.taxa), ncol = 2); rownames(sau.estimates) <- sau.taxa; colnames(sau.estimates) <- c("MR", "QE")  
for(i in sau.taxa) {  
 taxon <- which(bs.data$Taxon == i)  
 if(length(taxon) > 1) row <- taxon[1]  
 else row <- taxon  
 sau.estimates[i, "MR"] <- bs.data[row, "MR"]  
 sau.estimates[i, "QE"] <- bs.data[row, "QE"]  
}  
mean(sau.estimates[,1]-sau.estimates[,2], na.rm = TRUE)

## [1] -3498380

kruskal.test(mass~model, data = data.frame(mass = c(as.vector(sau.estimates)), model = as.factor(c(rep("MR", length(sau.taxa)), rep("QE", length(sau.taxa))))))

##   
## Kruskal-Wallis rank sum test  
##   
## data: mass by model  
## Kruskal-Wallis chi-squared = 0.62027, df = 1, p-value = 0.4309

## Custom Functions

A variety of custom functions were generated to run analyses and generate plots. Most relevant, however, is ‘MASSunite’, which is now part of MASSTIMATE v2.0 and carries out the comparative framework proposed in this study and used to generate Figs 10–12, 14, and 15. A number of additional mass estimate equations (MCF2004, CF2004, OH2012) are here included and updated in MASSTIMATE v2.0.

#extant bivariate plot  
ols.reg <- function(xvar,yvar, ...) {  
 plot(xvar,yvar,xlab = "Log Stylopodial Circumference",ylab = "Log Body Mass",cex = 0.5,pch = 4,col = "gray",  
 xlim = range(log10(c(bs.data$FC,bs.data$HCFC, extant.data$HCFC)),na.rm = TRUE),  
 ylim = range(log10(c(bs.data$BM.g, extant.data$BM)),na.rm = TRUE),  
 ...)  
 fit <- lm(yvar~xvar) #OLS regression analysis  
 abline(fit,col = "black")  
 #for intervals  
 seq <- seq(from = 1,to = 4,by = 0.01)  
 #prediction lines  
 ols.pred <- predict(fit,newdata = data.frame(xvar = seq),interval = "prediction")  
 lines(seq,ols.pred[,3],col = "blue",cex = 2,lty = 2)  
 lines(seq,ols.pred[,2],col = "blue",cex = 2,lty = 2)  
 #confidence lines  
 ols.conf <- predict(fit,newdata = data.frame(xvar = seq),interval = "confidence")  
 lines(seq,ols.conf[,3],col = "red",cex = 2,lty = 1)  
 lines(seq,ols.conf[,2],col = "red",cex = 2,lty = 1)  
 return(list(regression = fit,  
 prediction = ols.pred,  
 confidence = ols.conf))  
}  
  
#extant quadratic plot  
quad.reg <- function(xvar,yvar, ...) {  
 plot(xvar,yvar,xlab = "Log Stylopodial Circumference",ylab = "Log Body Mass",cex = 0.5,pch = 4,col = "gray",  
 xlim = range(log10(c(bs.data$FC,bs.data$HCFC)),na.rm = TRUE),  
 ylim = range(log10(bs.data$BM.g),na.rm = TRUE), ...)  
 fit <- lm(yvar~xvar+I(xvar^2)) #OLS regression analysis  
 curve(fit$coefficients[3]\*x^2+fit$coefficients[2]\*x+fit$coefficients[1],add = TRUE,from = 1,to = 4)  
 #for intervals  
 seq <- seq(from = 1,to = 4,by = 0.01)  
 #prediction lines  
 quad.pred <- predict(fit,newdata = data.frame(xvar = seq),interval = "prediction")  
 lines(seq,quad.pred[,3],col = "blue",cex = 2,lty = 2)  
 lines(seq,quad.pred[,2],col = "blue",cex = 2,lty = 2)  
 #confidence lines  
 quad.conf <- predict(fit,newdata = data.frame(xvar = seq),interval = "confidence")  
 lines(seq,quad.conf[,3],col = "red",cex = 2,lty = 1)  
 lines(seq,quad.conf[,2],col = "red",cex = 2,lty = 1)  
 return(list(regression = fit,  
 prediction = quad.pred,  
 confidence = quad.conf))  
}  
  
#mass comparison plot  
mass.plots <- function(data,grp,pch,col,type, eqn = "bivariate") {  
 ###dinos on plot  
 close.screen(all.screens = TRUE)  
 split.screen(c(2,1))  
 #split.screen(c(1,2),1)  
 l.grp <- length(levels(grp))  
 par(mar = c(5,4,1,1))  
 if(eqn == "bivariate") {  
 extant.ols <- ols.reg(extant.data$log.HCFC,extant.data$log.BM)  
 BM <- data$QE  
 logBM <- data$logQE  
 }  
 if(eqn == "quadratic") {  
 extant.ols <- quad.reg(extant.data$log.HCFC,extant.data$log.BM)  
 BM <- data$qQE  
 logBM <- data$logqQE  
 }  
 bip <- which(data$Gait == "bip")  
 quad <- which(data$Gait == "quad")  
 points(log10(data$HCFC[quad]),log10(data$BM.g[quad]),cex = 0.75,pch = pch[grp[quad]],col = col[grp[quad]])  
 points(log10(data$FC[bip]\*sqrt(2)),log10(data$BM.g[bip]),cex = 0.75,pch = pch[grp[bip]],col = col[grp[bip]])  
 #legend("bottomright",legend = levels(grp),pch = pch,col = col,cex = 0.75,bty = "n")  
 legend("topleft",legend = c('Extant Data','Extant Line','95% Confidence Intervals','95% Prediction Intervals'),  
 title = "Extant Scaling Legend",col = c('gray','black','red','blue'),lty = c(NA,1,1,2),pch = c(4,NA,NA,NA),cex = 0.65)  
 ###residual plot  
 screen(2)  
 dinos.res <- log10(data$BM.g)-logBM  
 par(mar = c(5,4,1,1))  
 plot(fitted(extant.ols),residuals(extant.ols),xlim = range(logBM,na.rm = TRUE),ylim = range(dinos.res,na.rm = TRUE),cex = 0.5,pch = 4,col = "gray",  
 xlab = "Fitted Values",  
 ylab = "Residuals")  
 points(logBM[quad],dinos.res[quad],cex = 0.75,pch = pch[grp[quad]],col = col[grp[quad]])  
 points(logBM[bip],dinos.res[bip],cex = 0.75,pch = pch[grp[bip]],col = col[grp[bip]])  
 lines(extant.ols$prediction[,1],extant.ols$prediction[,3]-extant.ols$prediction[,1],lty = 2, col = "blue") #upper prediction interval  
 lines(extant.ols$prediction[,1],extant.ols$prediction[,2]-extant.ols$prediction[,1],lty = 2, col = "blue") #lower prediction interval  
 abline(h = 0)  
 ###residual boxplot  
 close.screen(all.screens = TRUE)  
 split.screen(c(2,1))  
 l.grp <- levels(grp)  
 labels <- l.grp  
 if(type == "clade") {  
 labels <- abbreviate(l.grp)  
 }  
 par(mar = c(5,4,1,1))  
 boxplot(dinos.res~grp,xaxt = "n", ylab = "Residuals",col = col)  
 axis(1,at = 1:length(l.grp),labels = labels)  
 box.xaxis <- seq(from = 1,to = 301,by = 301/length(l.grp))  
 lines(1:length(box.xaxis),extant.ols$prediction[box.xaxis,3]-extant.ols$prediction[box.xaxis,1],lty = 2, col = "blue") #upper prediction interval  
 lines(1:length(box.xaxis),extant.ols$prediction[box.xaxis,2]-extant.ols$prediction[box.xaxis,1],lty = 2, col = "blue") #lower prediction interval  
 ###Absolute PPE boxplot  
 screen(2)  
 dinos.ppe <- ppe(true = bs.data$BM.g,pred = BM)  
 par(mar = c(5,4,1,1))  
 plot(grp,dinos.ppe$ppe.list,xaxt = "n",ylab = "Absolute Percent Prediction Error",col = col)  
 axis(1,at = 1:length(l.grp),labels = labels)  
 live.ppe <- ppe(true = extant.data$BM,pred = 10^(fitted(extant.ols$regression)))  
 abline(a = live.ppe$mean,b = 0,lty = 2) #average percent prediction error in extant data set  
 ###PPE boxplot  
 #split.screen(c(2,1))  
 close.screen(all.screens = TRUE)  
 #plot(grp,dinos.ppe,xaxt = "n",ylab = "Percent Prediction Error",,col = col)  
 #axis(1,at = 1:length(l.grp),labels = labels)  
 #abline(a = 25.63,b = 0,lty = 2)  
 #abline(a = -25.63,b = 0,lty = 2)  
 ###residual/PPE matrix  
 mat <- matrix(nrow = length(pch)+1,ncol = 7)  
 colnames(mat) <- c('N','mean res','sd res','res.p','mean PPE','sd PPE','PPE.p')  
 rownames(mat) <- c(levels(grp),'All')  
 for(i in 1:length(levels(grp))) {  
 x <- which(grp == levels(grp)[i])  
 mat[i,1:3] <- c(length(na.omit(dinos.res[x])),  
 round(mean(dinos.res[x],na.rm = TRUE),4),  
 round(sd(dinos.res[x],na.rm = TRUE),4))  
   
 if(length(na.omit(dinos.res[x])) >= 5) {  
 res.ttest <- t.test(dinos.res[x])  
 if(res.ttest$p.value<0.05) mat[i,4] <- "\*"  
 if(res.ttest$p.value<0.01) mat[i,4] <- "\*\*"  
 if(res.ttest$p.value<0.001) mat[i,4] <- "\*\*\*"  
 if(res.ttest$p.value>0.05) mat[i,4] <- "-"  
 ppe.ttest <- t.test(dinos.ppe$ppe.list[x],live.ppe$ppe.list)  
 if(ppe.ttest$p.value<0.05) mat[i,7] <- "\*"  
 if(ppe.ttest$p.value<0.01) mat[i,7] <- "\*\*"  
 if(ppe.ttest$p.value<0.001) mat[i,7] <- "\*\*\*"  
 if(ppe.ttest$p.value>0.05) mat[i,7] <- "-"  
 }  
 mat[i,5:6] <- c(round(mean(dinos.ppe$ppe.list[x],na.rm = TRUE),2),  
 round(sd(dinos.ppe$ppe.list[x],na.rm = TRUE),2))  
 }  
 mat[length(levels(grp))+1,1] <- length(na.omit(dinos.res))  
 mat[length(levels(grp))+1,2] <- round(mean(dinos.res,na.rm = TRUE),4)  
 mat[length(levels(grp))+1,3] <- round(sd(dinos.res,na.rm = TRUE),4)  
 res.ttest <- t.test(dinos.res)  
 if(res.ttest$p.value<0.05) mat[length(levels(grp))+1,4] <- "\*"  
 if(res.ttest$p.value<0.01) mat[length(levels(grp))+1,4] <- "\*\*"  
 if(res.ttest$p.value<0.001) mat[length(levels(grp))+1,4] <- "\*\*\*"  
 if(res.ttest$p.value>0.05) mat[length(levels(grp))+1,4] <- "-"  
 mat[length(levels(grp))+1,5] <- round(dinos.ppe$mean,2)  
 mat[length(levels(grp))+1,6] <- round(dinos.ppe$st.dev,2)  
 ppe.ttest <- t.test(dinos.ppe$ppe.list,live.ppe$ppe.list)  
 if(ppe.ttest$p.value<0.05) mat[length(levels(grp))+1,7] <- "\*"  
 if(ppe.ttest$p.value<0.01) mat[length(levels(grp))+1,7] <- "\*\*"  
 if(ppe.ttest$p.value<0.001) mat[length(levels(grp))+1,7] <- "\*\*\*"  
 if(ppe.ttest$p.value>0.05) mat[length(levels(grp))+1,7] <- "-"  
 return(list(Extant.OLS = extant.ols,  
 Proportion.in.95Pi = length(which(abs(dinos.res)<0.2659))/length(na.omit(dinos.res)),  
 Proportion.above.95Pi = length(which(dinos.res > 0.2659))/length(na.omit(dinos.res)),  
 Proportion.below.95Pi = length(which(dinos.res < (-0.2659)))/length(na.omit(dinos.res)),  
 Test.Results = as.data.frame(mat)))  
}  
  
# unified comparative framework proposed in this study  
MASSunite <- function(HFC = NULL, HC, FC, BM, ES.type = "QE", eqn, names = NULL, hybrid.data = NULL, pch, col, line = FALSE, ...) {  
 par(mar = c(5, 4, 1, 1), mfrow = c(1, 2))  
   
 if(ES.type == "QE") ols.reg(extant.data$log.HCFC, extant.data$log.BM, ...)  
 if(ES.type == "qQE") quad.reg(extant.data$log.HCFC, extant.data$log.BM, ...)  
   
 if(eqn == "biped") {  
 HFC <- FC \* sqrt(2) #apply corrections factor  
 est.BM <- cQE(FC)  
 if(!is.null(hybrid.data)) {  
 est.hyb <- cQE(hybrid.data[,1])  
 hyb.HFC <- hybrid.data[,1] \* sqrt(2)  
 }  
 }  
 if (eqn == "quadruped") {  
 if(is.null(HFC)) HFC <- HC + FC  
 est.BM <- QE(HFC)  
 if(!is.null(hybrid.data)) {  
 hyb.HFC <- hybrid.data[,1]  
 est.hyb <- QE(hybrid.data[,1])  
 }  
 }  
 points(log10(HFC), log10(BM), pch = pch, col = col)  
 if(line) {  
 mod.line <- lm(log10(BM)~log10(HFC))  
 abline(mod.line, col = col)  
 }  
 #if(!is.null(names)) text(log10(HFC), log10(BM), labels = names, pos = 4)  
   
 #deviation metrics  
 res.BM <- log10(BM)-est.BM[,1]  
 ppe.BM <- ppe(BM, est.BM[,2])  
   
 if(!is.null(hybrid.data)) {  
 points(log10(hyb.HFC), log10(hybrid.data[,2]), pch = 1)  
 hyb.res.BM <- log10(hybrid.data[,2])-est.hyb[,1]  
 par(mar = c(5, 4, 1, 1))  
 plot(1:2, 1:2, ylim = range(dinos.res, na.rm = TRUE), xlim = c(0, 3), type = "n")  
 abline(a = 0.2659,b = 0,lty = 2,col = "blue")  
 abline(a = -0.2659,b = 0,lty = 2,col = "blue")  
 boxplot(res.BM, add = TRUE, at = 1, col = col, ylab = "Residual Deviations")  
 boxplot(hyb.res.BM, add = TRUE, at = 2, ylab = "")  
 if(!is.null(names)) text(rep(2, length(hyb.res.BM)), hyb.res.BM, label = hybrid.data$Taxon)  
 }  
 else {  
 par(mar = c(5, 4, 1, 1))  
 if(length(res.BM > 1)) boxplot(res.BM, ylim = range(dinos.res, na.rm = TRUE), col = col, ylab = "Residual Deviations")  
 else plot(1, res.BM, ylim = range(dinos.res, na.rm = TRUE), pch = 19, col = col, ylab = "Residual Deviations")  
 abline(a = 0.2659,b = 0,lty = 2,col = "blue")  
 abline(a = -0.2659,b = 0,lty = 2,col = "blue")  
 if(!is.null(names)) text(rep(1, length(res.BM)), res.BM, labels = names)  
 }  
 outs <- which(res.BM > 0.2659 | res.BM < (-0.2659))  
 if(length(outs) >= 1 & line == TRUE) {  
 res <- list(Residual = res.BM, mean.Residual = mean(res.BM, na.rm = TRUE), PPE = ppe.BM, Inferred.Line = mod.line, Outliers = outs, Caution = warning("One or more of the models are inconsistent with ES expectations"))  
 }  
 if(length(outs) >= 1 & line != TRUE) {  
 res <- list(Residual = res.BM, mean.Residual = mean(res.BM, na.rm = TRUE), PPE = ppe.BM, Outliers = outs, Caution = warning("One or more of the models are inconsistent with ES expectations"))  
 }  
 if(length(outs) < 1 & line == TRUE) {  
 res <- list(Residual = res.BM, mean.Residual = mean(res.BM, na.rm = TRUE), PPE = ppe.BM, Inferred.Line = mod.line)  
 }  
 if(length(outs) < 1 & line != TRUE) {  
 res <- list(Residual = res.BM, mean.Residual = mean(res.BM, na.rm = TRUE), PPE = ppe.BM)  
 }  
 return(res)  
}  
  
  
#Updated Christiansen & Fariña (2004) function  
CF2004 <- function(X, X2 = NULL, eqn, data = NULL) {  
 if(eqn == "FL") {  
 log.CF2004 <- 3.222 \* log10(X) - 6.288  
 CF2004 <- round((10^(log.CF2004)) \* 1000, 2)  
 return(cbind(data, log.CF2004, CF2004))  
 }  
 if(eqn == "FC") {  
 log.CF2004 <- 2.738 \* log10(X) - 3.607  
 CF2004 <- round((10^log.CF2004) \* 1000, 2)  
 return(cbind(data, log.CF2004, CF2004))  
 }  
 if(eqn == "FAP") {  
 log.CF2004 <- 2.641 \* log10(X) - 2.028  
 CF2004 <- round((10^log.CF2004) \* 1000, 2)  
 return(cbind(data, log.CF2004, CF2004))  
 }  
 if(eqn == "FML") {  
 log.CF2004 <- 2.377 \* log10(X) - 2.284  
 CF2004 <- round((10^log.CF2004) \* 1000, 2)  
 return(cbind(data, log.CF2004, CF2004))  
 }  
 if(eqn == "TC") {  
 log.CF2004 <- 2.611 \* log10(X) - 3.135  
 CF2004 <- round((10^log.CF2004) \* 1000, 2)  
 return(cbind(data, log.CF2004, CF2004))  
 }  
 if(eqn == "TdistML") {  
 log.CF2004 <- 2.337 \* log10(X) - 2.099  
 CF2004 <- round((10^log.CF2004) \* 1000, 2)  
 return(cbind(data, log.CF2004, CF2004))  
 }  
 if(eqn == "FidistAP") {  
 log.CF2004 <- 2.787 \* log10(X) - 2.905  
 CF2004 <- round((10^log.CF2004) \* 1000, 2)  
 return(cbind(data, log.CF2004, CF2004))  
 }  
 if(eqn == "FC+FL") {  
 log.CF2004 <- 1.030 \* log10(X) + 2.012 \* log10(X2) - 5.285 #X=FC, X2=FL  
 CF2004 <- round((10^log.CF2004) \* 1000, 2)  
 return(cbind(data, log.CF2004, CF2004))  
 }  
 if(eqn == "FML+FL") {  
 log.CF2004 <- 0.431 \* log10(X) + 2.714 \* log10(X2) - 5.656 #X=FML, X2=FL  
 CF2004 <- round((10^log.CF2004) \* 1000, 2)  
 return(cbind(data, log.CF2004, CF2004))  
 }   
 if(eqn == "TC+FL") {  
 log.CF2004 <- 0.868 \* log10(X) + 2.159 \* log10(X2) - 5.263 #X=TC, X2=FL  
 CF2004 <- round((10^log.CF2004) \* 1000, 2)  
 return(cbind(data, log.CF2004, CF2004))  
 }   
 if(eqn == "TL+FC") {  
 log.CF2004 <- 0.715 \* log10(X) + 2.179 \* log10(X2) - 4.324 #X=TL, X2=FC  
 CF2004 <- round((10^log.CF2004) \* 1000, 2)  
 return(cbind(data, log.CF2004, CF2004))  
 }  
 if(eqn == "TC+FAP") {  
 log.CF2004 <- 1.461 \* log10(X) + 1.158 \* log10(X2) - 2.638 #X=TC, X2=FAD  
 CF2004 <- round((10^log.CF2004) \* 1000, 2)  
 return(cbind(data, log.CF2004, CF2004))  
 }  
}  
  
#Updated Mazzetta et al. (2004) function  
MCF2004 <- function(X, eqn, data = NULL) {  
 if(eqn == "FL") {  
 log.MCF2004 <- 3.195 \* log10(X) - 5.983  
 MCF2004 <- round((10^log.MCF2004) \* 1000, 2)  
 return(cbind(data, log.MCF2004, MCF2004))  
 }  
 if(eqn == "FL\*") {  
 log.MCF2004 <- 2.838 \* log10(X) - 4.864  
 MCF2004 <- round((10^log.MCF2004) \* 1000, 2)  
 return(cbind(data, log.MCF2004, MCF2004))  
 }  
 if(eqn == "FC") {  
 log.MCF2004 <- 2.955 \* log10(X) - 4.166  
 MCF2004 <- round((10^log.MCF2004) \* 1000, 2)  
 return(cbind(data, log.MCF2004, MCF2004))  
 }  
 if(eqn == "TL") {  
 log.MCF2004 <- 3.876 \* log10(X) - 7.342  
 MCF2004 <- round((10^log.MCF2004) \* 1000, 2)  
 return(cbind(data, log.MCF2004, MCF2004))  
 }  
 if(eqn == "TC") {  
 log.MCF2004 <- 3.288 \* log10(X) - 4.507  
 MCF2004 <- round((10^log.MCF2004) \* 1000, 2)  
 return(cbind(data, log.MCF2004, MCF2004))  
 }  
}  
  
# Equations of O'Gorman & Hone (2012)  
OH2012 <- function(FL, eqn, data=NULL) {  
 if(eqn == "theropods") {  
 log.OH2012 <- 3.1854 \* log10(FL) - 3.184  
 OH2012 <- round((10^log.OH2012), 2)  
 return(cbind(data, log.OH2012, OH2012))  
 }  
 if(eqn == "sauropods") {  
 log.OH2012 <- 2.3459 \* log10(FL) - 0.2935  
 OH2012 <- round((10^log.OH2012), 2)  
 return(cbind(data, log.OH2012, OH2012))  
 }  
 if(eqn == "ornithischians") {  
 log.OH2012 <- 3.0587 \* log10(FL) - 2.7042  
 OH2012 <- round((10^log.OH2012), 2)  
 return(cbind(data, log.OH2012, OH2012))  
 }  
}

# Main Analyses

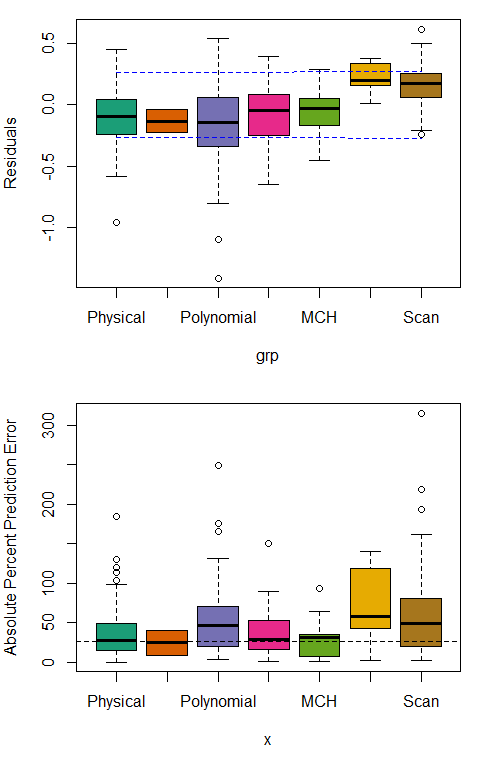
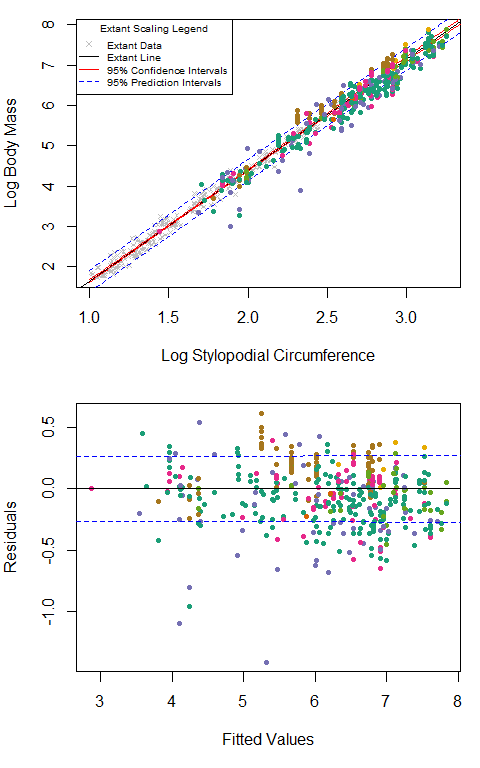
The following are the principal analyses and results presented in the manuscript. Results grouped by clade and methods are presented in Figs 6, 7, and 13, along with Tables 1 and 5, but additional results are provided here for other grouping factors, including by year, ontogeny, matching category, and gait. Size and year are also assessed in a continuous framework, see Fig. 8 and Table 3. Finally, all factors are assessed within a non-parametric ANOVA, applied to the raw and absolute residual deviations between ES and VD mass estimates; these results are shown in Table 2.

## By Clade

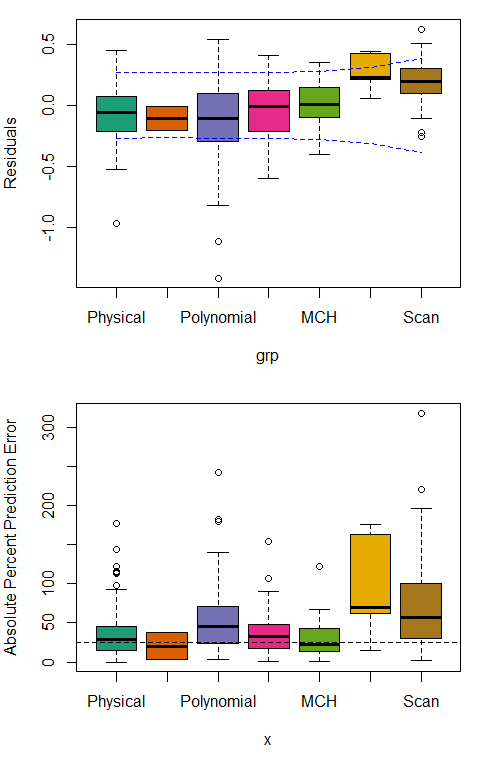
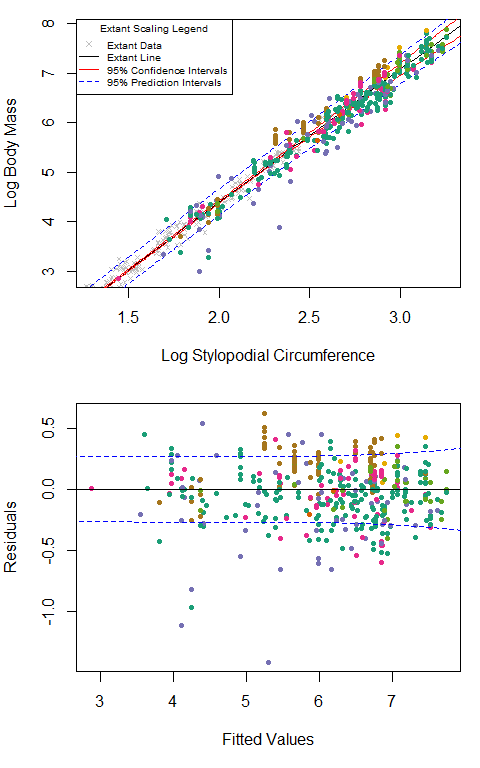
par(mar = c(2,4,1,1))  
clade.levels <- levels(as.factor(bs.data$Clade.2))  
clade.pch <- rep(19, length(clade.levels))  
clade.2.sort <- factor(bs.data$Clade.2, names(sort(tapply(bs.data$logQE, as.factor(bs.data$Clade.2), mean, na.rm = TRUE)))) #sort factor vector for plotting  
pdf("by clade.pdf", width = 7.0866142, height = 12, useDingbats = TRUE)  
#linear model, low taxonomy  
clade.res.ols <- mass.plots(bs.data,clade.2.sort,pch = clade.pch,col = clade.2.col[levels(clade.2.sort)],"clade", eqn = "bivariate")  
#quadratic model, low taxonomy  
clade.res.quad <- mass.plots(bs.data,clade.2.sort,pch = clade.pch,col = clade.2.col[levels(clade.2.sort)],"clade", eqn = "quadratic")  
#linear mode, high taxonomy  
clade.1.res <- mass.plots(bs.data, as.factor(bs.data$Clade.1), pch = c(19, 19, 19), col = c("#006e8d","#ad1457","#263802"), "clade")  
#quadrataic model, high taxonomy  
clade.1.res.quad <- mass.plots(bs.data, as.factor(bs.data$Clade.1), pch = c(19, 19, 19), col = c("#006e8d","#ad1457","#263802"), "clade", eqn = "quadratic")  
#dev.off()  
  
##outlier deviations (outside 95% PI) - NOT RUN  
#outs <- which(dinos.res > 0.2659 | dinos.res < (-0.2659))  
#out.data <- data.frame(bs.data[outs,], Absolute.Residuals = abs(dinos.res[outs]))  
#tapply(out.data$Method, out.data$Method, length)  
#tapply(out.data$mature, out.data$mature, length)  
#tapply(out.data$EBM.ref, out.data$EBM.ref, length)  
#write.csv(out.data[order(-(out.data$Absolute.Residuals)),], file = "Results/outliers.csv")

## By VD Methods

##methods  
meth.pch <- c(19,19,19,19,19,19,19)  
meth.sort <- factor(bs.data$Method, method.order)  
#pdf("Results/by method.pdf", width = 7.0866142, height = 12, useDingbats = TRUE)  
meth.res <- mass.plots(bs.data,meth.sort,pch = meth.pch,col = brewer.pal(length(meth.pch), name = "Dark2"),type = "method")



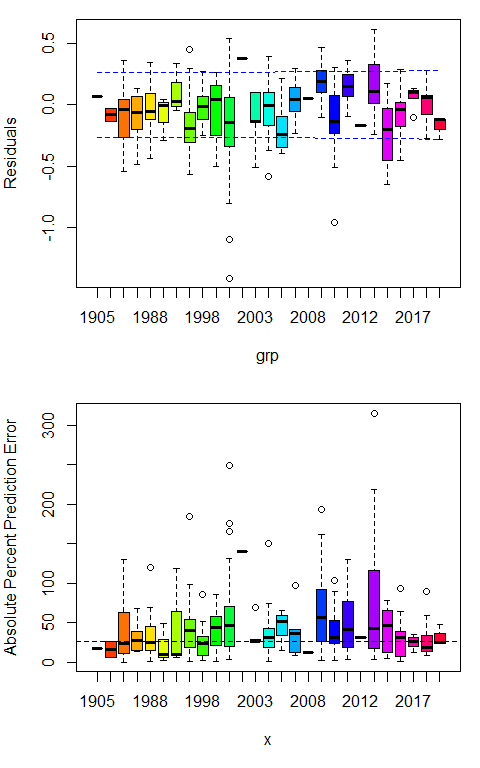
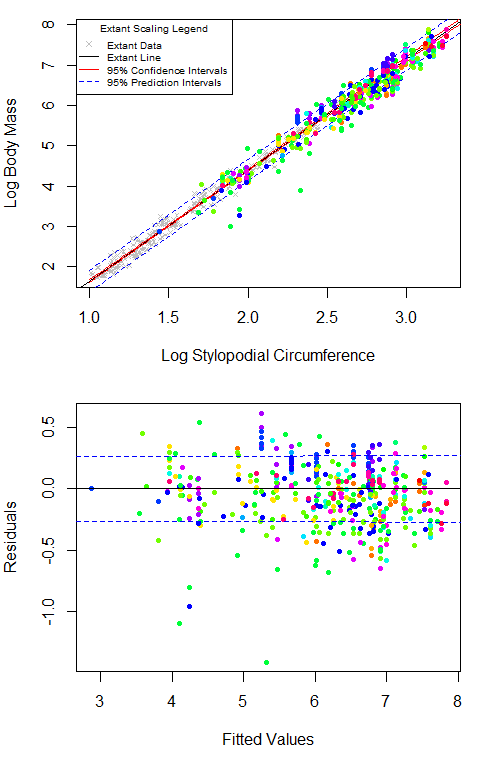
meth.res.quad <- mass.plots(bs.data,meth.sort,pch = meth.pch,col = brewer.pal(length(meth.pch), name = "Dark2"),type = "method", eqn = "quadratic")



#dev.off()

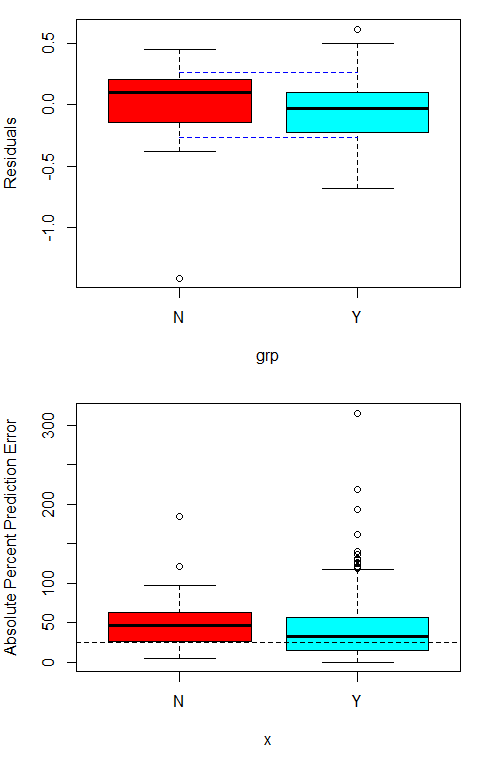
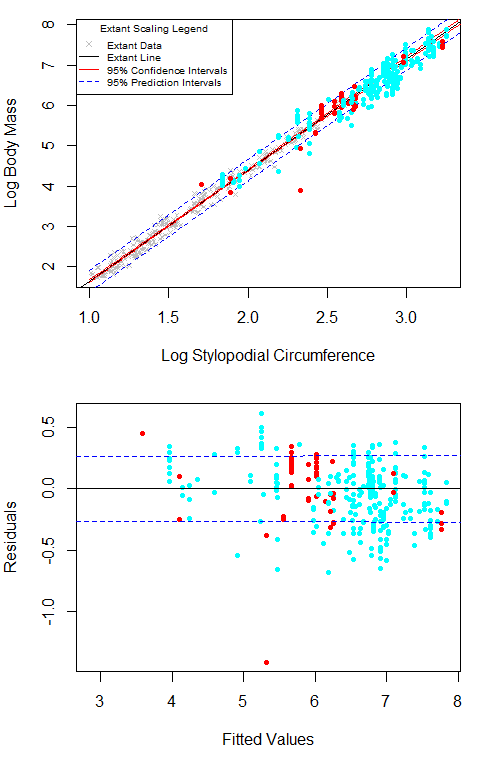
## By Year VD model was generated

year.pch <- rep(19,27)  
year.res <- mass.plots(bs.data,as.factor(bs.data$EBM.yr),year.pch,rainbow(length(year.pch)),"Year")



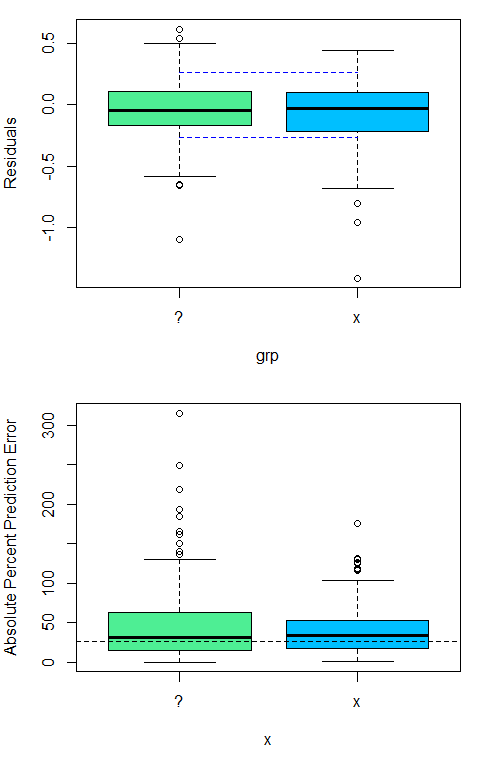
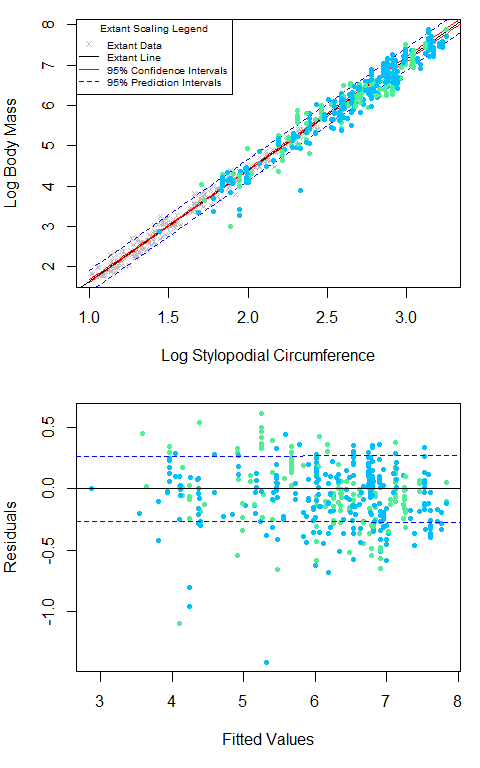
## By Ontogeny

ontog.pch <- c(19,19)  
ontog.res <- mass.plots(bs.data,as.factor(bs.data$mature),ontog.pch,rainbow(length(ontog.pch)),"Maturity")



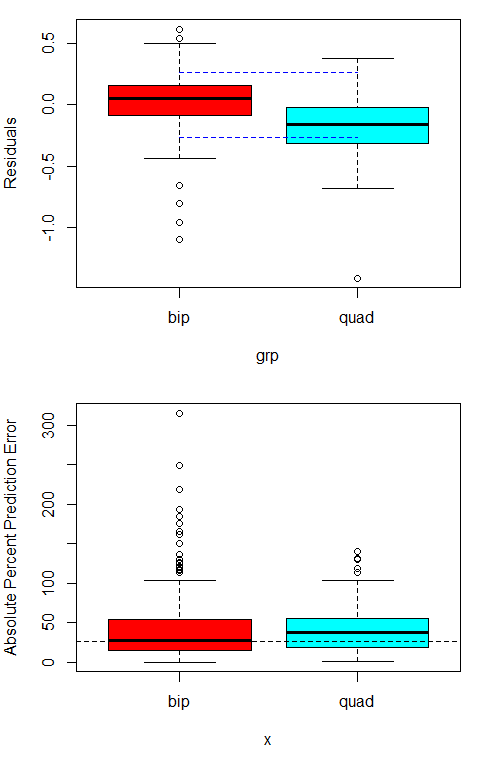
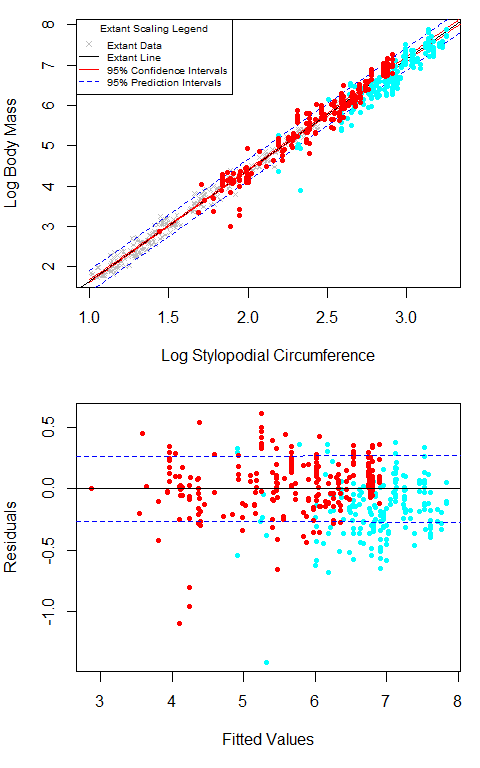
## By Matching Category (full data set vs. Perfect Match)

#only runs with the complete data set  
match.pch <- c(19,19)  
match.res <- mass.plots(bs.data,as.factor(bs.data$Match),match.pch,c("seagreen2", "deepskyblue"),"Perfect Match")



## By Gait

gait.pch <- c(19,19)  
gait.res <- mass.plots(bs.data,as.factor(bs.data$Gait),gait.pch,rainbow(length(gait.pch)),"Perfect Match")



## Relationship with Size and Year of Publication

Size and year linear results are used to make Fig. 8 and statistics are tabulated in Table 3.

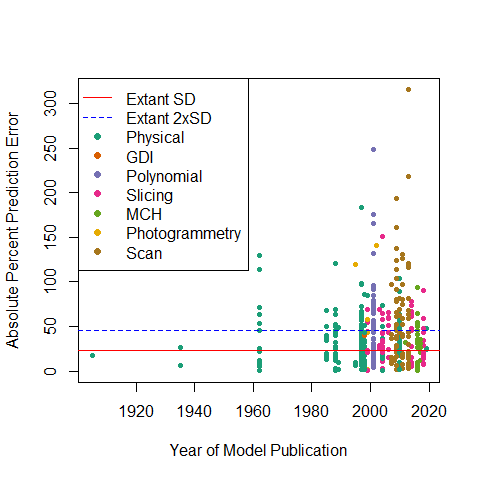
### Relationship with Size

## Rank Correlation  
pdf("Figure 8AB.pdf", width = 7.0866142, height = 12)  
par(mfrow = c(2,1), mar = c(5,4,1,1))  
# residuals  
plot(bs.data$logQE, dinos.res, xlab = "Fitted Values (log10 g)", ylab = "Residuals", pch = 19, col = brewer.pal(length(meth.pch), name = "Dark2")[meth.sort])  
abline(lm(dinos.res~bs.data$logQE))  
size.spear.all <- cor.test(bs.data$logQE, dinos.res, method = "spearman")  
size.spear.meth <- matrix(nrow = length(meth.pch), ncol = 3); rownames(size.spear.meth) <- levels(meth.sort); colnames(size.spear.meth) <- c("N", "Rho", "p-value")  
for(i in 1:length(meth.pch)) {  
 meth <- which(bs.data$Method == levels(meth.sort)[i])  
 size.spear <- cor.test(bs.data$logQE[meth], dinos.res[meth], method = "spearman")  
 size.spear.meth[i,] <- c(length(meth), size.spear$estimate, size.spear$p.value)  
 abline(lm(dinos.res[meth]~bs.data$logQE[meth]), col = brewer.pal(length(meth.pch), name = "Dark2")[i])  
}  
#PPE  
plot(bs.data$logQE, dinos.ppe$ppe.list, xlab = "Fitted Values (log10 g)", ylab = "Absolute PPE", pch = 19, col = brewer.pal(length(meth.pch), name = "Dark2")[meth.sort])  
 abline(lm(dinos.ppe$ppe.list~bs.data$logQE))  
size.spear.all.2 <- cor.test(bs.data$logQE, dinos.ppe$ppe.list, method = "spearman")  
size.spear.meth.2 <- matrix(nrow = length(meth.pch), ncol = 3); rownames(size.spear.meth.2) <- levels(meth.sort); colnames(size.spear.meth.2) <- c("N", "Rho", "p-value")  
for(i in 1:length(meth.pch)) {  
 meth <- which(bs.data$Method == levels(meth.sort)[i])  
 size.spear <- cor.test(bs.data$logQE[meth], dinos.ppe$ppe.list[meth], method = "spearman")  
 size.spear.meth.2[i,] <- c(length(meth), size.spear$estimate, size.spear$p.value)  
 abline(lm(dinos.ppe$ppe.list[meth]~bs.data$logQE[meth]), col = brewer.pal(length(meth.pch), name = "Dark2")[i])  
}  
dev.off()

## png   
## 2

### Relationship with Year of Publication

##scatterplot - ppe  
plot(bs.data$EBM.yr,dinos.ppe$ppe.list,cex = 0.75,ylab = "Absolute Percent Prediction Error", xlab = "Year of Model Publication",pch = 19, col = brewer.pal(length(meth.pch), name = "Dark2")[meth.sort])  
extant.ppe <- ppe(true = extant.data$BM, pred = 10^(fitted(clade.res.ols$Extant.OLS$regression)))  
abline(h = c(extant.ppe$st.dev, extant.ppe$st.dev\*2), lty = 1:2, col = c("red", "blue"))  
legend("topleft", legend = c("Extant SD", "Extant 2xSD", levels(meth.sort)), col = c("red", "blue", brewer.pal(length(meth.pch), name = "Dark2")), lty = c(1, 2, rep(NA, 7)), pch = c(NA, NA, meth.pch))

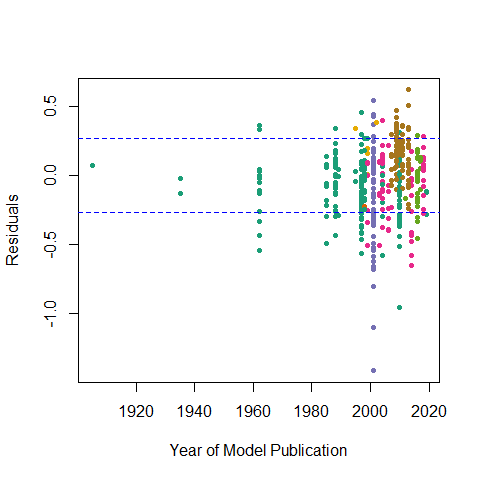


cor.test(bs.data$EBM.yr,dinos.ppe$ppe.list, method = "spearman")

## Warning in cor.test.default(bs.data$EBM.yr, dinos.ppe$ppe.list, method =  
## "spearman"): Cannot compute exact p-value with ties

##   
## Spearman's rank correlation rho  
##   
## data: bs.data$EBM.yr and dinos.ppe$ppe.list  
## S = 13046356, p-value = 0.5469  
## alternative hypothesis: true rho is not equal to 0  
## sample estimates:  
## rho   
## 0.02906189

##scatterplot - residual  
plot(bs.data$EBM.yr,dinos.res,cex = 0.75,ylab = "Residuals", xlab = "Year of Model Publication", pch = 19, col = brewer.pal(length(meth.pch), name = "Dark2")[meth.sort])  
abline(a = 0.2659,b = 0,lty = 2,col = "blue")  
abline(a = -0.2659,b = 0,lty = 2,col = "blue")



cor.test(bs.data$EBM.yr,dinos.res, method = "spearman")

## Warning in cor.test.default(bs.data$EBM.yr, dinos.res, method = "spearman"):  
## Cannot compute exact p-value with ties

##   
## Spearman's rank correlation rho  
##   
## data: bs.data$EBM.yr and dinos.res  
## S = 11356699, p-value = 0.001247  
## alternative hypothesis: true rho is not equal to 0  
## sample estimates:  
## rho   
## 0.1548098

pdf("Figure 8CD.pdf", width = 7.0866142, height = 12)  
par(mfrow = c(2,1), mar = c(5,4,1,1))  
##log different - residuals  
x <- 2020-bs.data$EBM.yr #added a year so as to remove 0 values  
plot(log10(x),dinos.res,cex = 0.75,xlim = rev(range(log10(x))),ylab = "Residuals", xlab = "log Difference Between 2020 and Year of Model Publication", pch = 19, col = brewer.pal(length(meth.pch), name = "Dark2")[meth.sort])  
abline(a = 0.2659,b = 0,lty = 2,col = "blue")  
abline(a = -0.2659,b = 0,lty = 2,col = "blue")  
year.ols <- lm(dinos.res~log10(x)) #but fails parametric tests  
summary(year.ols)

##   
## Call:  
## lm(formula = dinos.res ~ log10(x))  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -1.34630 -0.15780 0.02271 0.15748 0.65141   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.03498 0.04118 0.850 0.3961   
## log10(x) -0.08040 0.03473 -2.315 0.0211 \*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.248 on 430 degrees of freedom  
## Multiple R-squared: 0.01231, Adjusted R-squared: 0.01001   
## F-statistic: 5.359 on 1 and 430 DF, p-value: 0.02108

abline(year.ols)  
##log different - PPE  
plot(log10(x),dinos.ppe$ppe.list,cex = 0.75,xlim = rev(range(log10(x))),ylab = "Absolute Percent Prediction Error", xlab = "log Difference Between 2020 and Year of Model Publication" ,pch = 19, col = brewer.pal(length(meth.pch), name = "Dark2")[meth.sort])  
for(i in 1:length(levels(bs.data$Method))) {  
 points(log10(x)[bs.data$Method == levels(bs.data$Method)[i]],dinos.ppe$ppe.list[bs.data$Method == levels(bs.data$Method)[i]],pch = 19, col = rainbow(n = length(levels(bs.data$Method)))[i])  
}  
abline(a = 25.63,b = 0,lty = 2)  
dev.off()

## png   
## 2

## Summary ANOVA

aov.data <- data.frame(Residuals = dinos.res,   
 Clade = as.factor(bs.data$Clade.2),   
 Method = as.factor(bs.data$Method),  
 Match = as.factor(bs.data$Match),  
 Gait = as.factor(bs.data$Gait),  
 Year = bs.data$EBM.yr,  
 Size = bs.data$logQE)  
anova <- procD.lm(Residuals~Clade+Method+Match+Gait+Year+Size, data = aov.data, SS.type = "II", print.progress = FALSE)  
summary(anova)

##   
## Analysis of Variance, using Residual Randomization  
## Permutation procedure: Randomization of null model residuals   
## Number of permutations: 1000   
## Estimation method: Ordinary Least Squares   
## Sums of Squares and Cross-products: Type II   
## Effect sizes (Z) based on F distributions  
##   
## Df SS MS Rsq F Z Pr(>F)   
## Clade 20 3.7836 0.18918 0.14133 4.9181 4.8160 0.001 \*\*  
## Method 6 2.1841 0.36401 0.08158 9.4632 3.6573 0.001 \*\*  
## Match 1 0.3128 0.31277 0.01168 8.1312 1.4862 0.009 \*\*  
## Gait 1 0.4835 0.48345 0.01806 12.5682 1.7348 0.003 \*\*  
## Year 1 0.0715 0.07153 0.00267 1.8594 0.8556 0.194   
## Size 1 0.0291 0.02909 0.00109 0.7562 0.4425 0.367   
## Residuals 401 15.4249 0.03847 0.57617   
## Total 431 26.7714   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Call: procD.lm(f1 = Residuals ~ Clade + Method + Match + Gait + Year +   
## Size, SS.type = "II", data = aov.data, print.progress = FALSE)

anova.abs <- procD.lm(abs(Residuals)~Clade+Method+Match+Gait+Year+Size, data = aov.data, SS.type = "II", print.progress = FALSE)  
summary(anova.abs)

##   
## Analysis of Variance, using Residual Randomization  
## Permutation procedure: Randomization of null model residuals   
## Number of permutations: 1000   
## Estimation method: Ordinary Least Squares   
## Sums of Squares and Cross-products: Type II   
## Effect sizes (Z) based on F distributions  
##   
## Df SS MS Rsq F Z Pr(>F)   
## Clade 20 1.5392 0.07696 0.12751 3.5686 3.8102 0.001 \*\*  
## Method 6 0.6697 0.11162 0.05548 5.1756 2.7000 0.006 \*\*  
## Match 1 0.0109 0.01086 0.00090 0.5033 0.2536 0.481   
## Gait 1 0.4008 0.40081 0.03320 18.5845 1.8468 0.001 \*\*  
## Year 1 0.0056 0.00560 0.00046 0.2598 -0.0753 0.617   
## Size 1 0.2501 0.25014 0.02072 11.5986 1.6840 0.003 \*\*  
## Residuals 401 8.6482 0.02157 0.71639   
## Total 431 12.0720   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Call: procD.lm(f1 = abs(Residuals) ~ Clade + Method + Match + Gait +   
## Year + Size, SS.type = "II", data = aov.data, print.progress = FALSE)

# Discussion Analyses

The following code uses the MASSunite function in the discussion of hybrid approaches and specific VD models. All plots are saved to the working directory.

## On the Utility of Hybrid Approaches

# Christiansen & Fariña (2004), theropods only  
pdf("Figure 10AB.pdf", width = 7, height = 3.75) #activate to save figures to pdf file  
thers <- which(benson.data$Clade == "Theropoda") #theropods in Benson et al. (2018)  
hybrid.CF2004 <- bs.data[which(bs.data$Hybrid.CF2004 == "x"), c("FC", "BM.g", "Taxon")]  
CF2004.VDs <- MASSunite(FC = hybrid.CF2004$FC, BM = hybrid.CF2004$BM.g, eqn = "biped", pch = 19, col = "black", line = TRUE)

## Warning in MASSunite(FC = hybrid.CF2004$FC, BM = hybrid.CF2004$BM.g, eqn =  
## "biped", : One or more of the models are inconsistent with ES expectations

#Femur length  
CF2004.FL <- MASSunite(FC = benson.data$FC[thers], BM = CF2004(benson.data$FL, eqn = "FL")[thers,2], eqn = "biped", hybrid.data = hybrid.CF2004, pch = 19, col = sma.col[1], line = TRUE)

## Warning in MASSunite(FC = benson.data$FC[thers], BM = CF2004(benson.data$FL, :  
## One or more of the models are inconsistent with ES expectations

#Femur circumference  
CF2004.FC <- MASSunite(FC = benson.data$FC[thers], BM = CF2004(benson.data$FC, eqn = "FC")[thers,2], eqn = "biped", hybrid.data = hybrid.CF2004, pch = 19, col = sma.col[2], line = TRUE)  
#Femur anterior-posterior diameter  
CF2004.FAP <- MASSunite(FC = benson.data$FC[thers], BM = CF2004(benson.data$FAP, eqn = "FAP")[thers,2], eqn = "biped", hybrid.data = hybrid.CF2004, pch = 19, col = sma.col[3], line = TRUE)

## Warning in MASSunite(FC = benson.data$FC[thers], BM = CF2004(benson.data$FAP, :  
## One or more of the models are inconsistent with ES expectations

#Femur medial-lateral diameter  
CF2004.FML <- MASSunite(FC = benson.data$FC[thers], BM = CF2004(benson.data$FML, eqn = "FML")[thers,2], eqn = "biped", hybrid.data = hybrid.CF2004, pch = 19, col = sma.col[4], line = TRUE)

## Warning in MASSunite(FC = benson.data$FC[thers], BM = CF2004(benson.data$FML, :  
## One or more of the models are inconsistent with ES expectations

#Tibia circumference  
CF2004.TC <- MASSunite(FC = benson.data$FC[thers], BM = CF2004(benson.data$TC, eqn = "TC")[thers,2], eqn = "biped", hybrid.data = hybrid.CF2004, pch = 19, col = sma.col[5], line = TRUE)

## Warning in MASSunite(FC = benson.data$FC[thers], BM = CF2004(benson.data$TC, :  
## One or more of the models are inconsistent with ES expectations

#Femur circumference + length  
CF2004.FC.FL <- MASSunite(FC = benson.data$FC[thers], BM = CF2004(benson.data$FC, benson.data$FL, eqn = "FC+FL")[thers,2], eqn = "biped", hybrid.data = hybrid.CF2004, pch = 19, col = sma.col[6], line = TRUE)

## Warning in MASSunite(FC = benson.data$FC[thers], BM = CF2004(benson.data$FC, :  
## One or more of the models are inconsistent with ES expectations

#Femur medial-lateral diameter + length  
CF2004.FML.FL <- MASSunite(FC = benson.data$FC[thers], BM = CF2004(benson.data$FML, benson.data$FL, eqn = "FML+FL")[thers,2], eqn = "biped", hybrid.data = hybrid.CF2004, pch = 19, col = sma.col[7], line = TRUE)

## Warning in MASSunite(FC = benson.data$FC[thers], BM = CF2004(benson.data$FML, :  
## One or more of the models are inconsistent with ES expectations

#Tibia circumference + femur length  
CF2004.TC.FL <- MASSunite(FC = benson.data$FC[thers], BM = CF2004(benson.data$TC, benson.data$FL, eqn = "TC+FL")[thers,2], eqn = "biped", hybrid.data = hybrid.CF2004, pch = 19, col = sma.col[8], line = TRUE)

## Warning in MASSunite(FC = benson.data$FC[thers], BM = CF2004(benson.data$TC, :  
## One or more of the models are inconsistent with ES expectations

#Tibia length + Femur circumference  
CF2004.TL.FC <- MASSunite(FC = benson.data$FC[thers], BM = CF2004(benson.data$TL\_TLE, benson.data$FC, eqn = "TL+FC")[thers,2], eqn = "biped", hybrid.data = hybrid.CF2004, pch = 19, col = sma.col[9], line = TRUE)  
#Tibia circumference + Femur anterior-posterior diameter  
CF2004.TC.FAP <- MASSunite(FC = benson.data$FC[thers], BM = CF2004(benson.data$TC, benson.data$FAP, eqn = "TC+FAP")[thers,2], eqn = "biped", hybrid.data = hybrid.CF2004, pch = 19, col = sma.col[10], line = TRUE)  
dev.off()

## png   
## 2

# Mazzetta et al. (2004), quadrupedal sauropodomorphs only  
pdf("Figure 10C.pdf", width = 7, height = 3.75)  
sauro <- which(benson.data$Clade == "Sauropodomorpha" & benson.data$Subclade != "Sauropodomorpha")  
hybrid.MCF2004 <- bs.data[which(bs.data$Hybrid.MCF2004 == "x"), c("HCFC", "BM.g", "Taxon")]  
MCF2004.VDs <- MASSunite(HFC = hybrid.MCF2004$HCFC, BM = hybrid.MCF2004$BM.g, eqn = "quadruped", pch = 19, col = "black", line = TRUE)

## Warning in MASSunite(HFC = hybrid.MCF2004$HCFC, BM = hybrid.MCF2004$BM.g, : One  
## or more of the models are inconsistent with ES expectations

#Femur length  
MCF2004.FL <- MASSunite(HC = benson.data$HC[sauro], FC = benson.data$FC[sauro], BM = MCF2004(benson.data$FL, eqn = "FL")[sauro,2], eqn = "quadruped", hybrid.data = hybrid.MCF2004, pch = 19, col = sma.col[11], line = TRUE)

## Warning in MASSunite(HC = benson.data$HC[sauro], FC = benson.data$FC[sauro], :  
## One or more of the models are inconsistent with ES expectations

#Femur length (modified by MCF2004)  
MCF2004.FL2 <- MASSunite(HC = benson.data$HC[sauro], FC = benson.data$FC[sauro], BM = MCF2004(benson.data$FL, eqn = "FL\*")[sauro,2], eqn = "quadruped", hybrid.data = hybrid.MCF2004, pch = 19, col = sma.col[12], line = TRUE)

## Warning in MASSunite(HC = benson.data$HC[sauro], FC = benson.data$FC[sauro], :  
## One or more of the models are inconsistent with ES expectations

#Femur circumference  
MCF2004.FC <- MASSunite(HC = benson.data$HC[sauro], FC = benson.data$FC[sauro], BM = MCF2004(benson.data$FC, eqn = "FC")[sauro,2], eqn = "quadruped", hybrid.data = hybrid.MCF2004, pch = 19, col = sma.col[13], line = TRUE)

## Warning in MASSunite(HC = benson.data$HC[sauro], FC = benson.data$FC[sauro], :  
## One or more of the models are inconsistent with ES expectations

#Tibia length  
# Note that Diamantinasaurus [entry 73 in the 'sauro' object] was,erroneously, given a femoral length of 80 mm in the Benson et al. (2018) data set. It thusly excluded from the following analysis.  
MCF2004.TL <- MASSunite(HC = benson.data$HC[sauro][-73], FC = benson.data$FC[sauro][-73], BM = MCF2004(benson.data$TL\_TLE, eqn = "TL")[sauro,2][-73], eqn = "quadruped", hybrid.data = hybrid.MCF2004, pch = 19, col = sma.col[14], line = TRUE)

## Warning in MASSunite(HC = benson.data$HC[sauro][-73], FC = benson.data$FC[sauro]  
## [-73], : One or more of the models are inconsistent with ES expectations

#Tibia circumference  
MCF2004.TC <- MASSunite(HC = benson.data$HC[sauro], FC = benson.data$FC[sauro], BM = MCF2004(benson.data$TC, eqn = "TC")[sauro,2], eqn = "quadruped", hybrid.data = hybrid.MCF2004, pch = 19, col = sma.col[15], line = TRUE)

## Warning in MASSunite(HC = benson.data$HC[sauro], FC = benson.data$FC[sauro], :  
## One or more of the models are inconsistent with ES expectations

dev.off()

## png   
## 2

# O'Gormen & Hone (2012)  
pdf("Figure 12.pdf", width = 7, height = 3.75)  
#Theropods  
hybrid.OH2012.thers <- bs.data[which(bs.data$Hybrid.OH2012 == "x" & bs.data$Clade.1 == "Theropoda"), c("FC", "BM.g", "Taxon")]  
OH2012.VDs.thers <- MASSunite(FC = hybrid.OH2012.thers$FC, BM = hybrid.OH2012.thers$BM.g, eqn = "biped", pch = 2, col = "black")

## Warning in MASSunite(FC = hybrid.OH2012.thers$FC, BM =  
## hybrid.OH2012.thers$BM.g, : One or more of the models are inconsistent with ES  
## expectations

OH2012.thers <- MASSunite(FC = benson.data$FC[thers], BM = OH2012(benson.data$FL, eqn = "theropods")[thers,2], eqn = "biped", pch = 19, col = clade.1.col[3], hybrid.data = hybrid.OH2012.thers, line = TRUE)

## Warning in MASSunite(FC = benson.data$FC[thers], BM = OH2012(benson.data$FL, :  
## One or more of the models are inconsistent with ES expectations

#Sauropods  
hybrid.OH2012.spds <- bs.data[which(bs.data$Hybrid.OH2012 == "x" & bs.data$Clade.1 == "Sauropodomorpha"), c("HCFC", "HC", "FC", "BM.g", "Gait", "Taxon")]  
##bipeds  
OH2012.VDs.spds.bips <- MASSunite(FC = hybrid.OH2012.spds$FC[hybrid.OH2012.spds$Gait == "bip"], BM = hybrid.OH2012.spds$BM.g[hybrid.OH2012.spds$Gait == "bip"], eqn = "biped", pch = 2, col = "black")

## Warning in MASSunite(FC = hybrid.OH2012.spds$FC[hybrid.OH2012.spds$Gait == : One  
## or more of the models are inconsistent with ES expectations

benson.spds.bips <- benson.data[which(benson.data$Clade == "Sauropodomorpha" & benson.data$Gait == "bip"),]  
OH2012.spds.bips <- MASSunite(FC = benson.spds.bips$FC, BM = OH2012(benson.spds.bips$FL, eqn = "sauropods")[,2], eqn = "biped", pch = 17, col = clade.1.col[2], hybrid.data = hybrid.OH2012.spds[which(hybrid.OH2012.spds$Gait == "bip"),c("FC", "BM.g")], line = TRUE)

## Warning in MASSunite(FC = benson.spds.bips$FC, BM =  
## OH2012(benson.spds.bips$FL, : One or more of the models are inconsistent with ES  
## expectations

##quadrupeds  
OH2012.VDs.spds.quads <- MASSunite(HFC = hybrid.OH2012.spds$HCFC[hybrid.OH2012.spds$Gait == "quad"], BM = hybrid.OH2012.spds$BM.g[hybrid.OH2012.spds$Gait == "quad"], eqn = "quadruped", pch = 2, col = "black")

## Warning in MASSunite(HFC = hybrid.OH2012.spds$HCFC[hybrid.OH2012.spds$Gait == :  
## One or more of the models are inconsistent with ES expectations

benson.spds.quads <- benson.data[which(benson.data$Clade == "Sauropodomorpha" & benson.data$Gait == "quad"),]  
OH2012.spds.quads <- MASSunite(HC = benson.spds.quads$HC, FC = benson.spds.quads$FC, BM = OH2012(benson.spds.quads$FL, eqn = "sauropods")[,2], eqn = "quadruped", pch = 19, col = clade.1.col[2], hybrid.data = hybrid.OH2012.spds[which(hybrid.OH2012.spds$Gait == "quad"),c("HCFC", "BM.g")], line = TRUE)

## Warning in MASSunite(HC = benson.spds.quads$HC, FC = benson.spds.quads$FC, : One  
## or more of the models are inconsistent with ES expectations

#Ornithischians  
hybrid.OH2012.ortns <- bs.data[which(bs.data$Hybrid.OH2012 == "x" & bs.data$Clade.1 == "Ornithischia"), c("HCFC", "HC", "FC", "BM.g", "Gait", "Taxon")]  
#bipeds  
OH2012.VDs.ortns.bips <- MASSunite(FC = hybrid.OH2012.ortns$FC[hybrid.OH2012.ortns$Gait == "bip"], BM = hybrid.OH2012.ortns$BM.g[hybrid.OH2012.ortns$Gait == "bip"], eqn = "biped", pch = 2, col = "black")

## Warning in MASSunite(FC = hybrid.OH2012.ortns$FC[hybrid.OH2012.ortns$Gait == :  
## One or more of the models are inconsistent with ES expectations

benson.ortns.bips <- benson.data[which(benson.data$Clade == "Ornithischia" & benson.data$Gait == "bip"),]  
OH2012.ortns.bips <- MASSunite(FC = benson.ortns.bips$FC, BM = OH2012(benson.ortns.bips$FL, eqn = "ornithischians")[,2], eqn = "biped", pch = 17, col = clade.1.col[1], hybrid.data = hybrid.OH2012.ortns[which(hybrid.OH2012.ortns$Gait == "bip"),c("FC", "BM.g")], line = TRUE)

## Warning in MASSunite(FC = benson.ortns.bips$FC, BM =  
## OH2012(benson.ortns.bips$FL, : One or more of the models are inconsistent with  
## ES expectations

#quadrupeds  
OH2012.VDs.ortns.quads <- MASSunite(HFC = hybrid.OH2012.ortns$HCFC[hybrid.OH2012.ortns$Gait == "quad"], BM = hybrid.OH2012.ortns$BM.g[hybrid.OH2012.ortns$Gait == "quad"], eqn = "quadruped", pch = 2, col = "black")

## Warning in MASSunite(HFC = hybrid.OH2012.ortns$HCFC[hybrid.OH2012.ortns$Gait  
## == : One or more of the models are inconsistent with ES expectations

benson.ortns.quads <- benson.data[which(benson.data$Clade == "Ornithischia" & benson.data$Gait == "quad"),]  
OH2012.ortns.quads <- MASSunite(HC = benson.ortns.quads$HC, FC = benson.ortns.quads$FC, BM = OH2012(benson.ortns.quads$FL, eqn = "ornithischians")[,2], eqn = "quadruped", pch = 19, col = clade.1.col[1], hybrid.data = hybrid.OH2012.ortns[which(hybrid.OH2012.ortns$Gait == "quad"),c("HCFC", "BM.g")], line = TRUE)

## Warning in MASSunite(HC = benson.ortns.quads$HC, FC = benson.ortns.quads$FC, :  
## One or more of the models are inconsistent with ES expectations

dev.off()

## png   
## 2

# Therrien & Henderson (2007)  
pdf("Figure 11.pdf", width = 7, height = 3.75)  
res.TH2007 <- read\_xlsx("Campione&Evans\_Appendix\_S1.xlsx", sheet = 4)  
hybrid.TH2007 <- bs.data[which(bs.data$Hybrid.TH2007 == "x"), c("FC", "BM.g", "Taxon")]  
  
TH2007.VDs <- MASSunite(FC = hybrid.TH2007$FC, BM = hybrid.TH2007$BM.g, eqn = "biped", pch = 19, col = "black", line = TRUE)  
  
TH2007.SK <- MASSunite(FC = res.TH2007$FC, BM = res.TH2007$BM.g, eqn = "biped", hybrid.data = hybrid.TH2007, pch = 19, col = sma.col[16], line = TRUE)

## Warning in MASSunite(FC = res.TH2007$FC, BM = res.TH2007$BM.g, eqn = "biped", :  
## One or more of the models are inconsistent with ES expectations

dev.off()

## png   
## 2

## Shifting Paradigms and the “Evolution” of Body Mass Estimates

### Historical VD Models

pdf("Figure 14.pdf", width = 7, height = 3.75)  
gregs <- bs.data[grep("greg", bs.data$EBM.ref, ignore.case = TRUE),]  
gregs.VD <- MASSunite(HFC = gregs$HCFC, BM = gregs$BM.g, eqn = "quadruped", pch = 19, col = brewer.pal(length(meth.pch), name = "Dark2")[1], line = FALSE)

## Warning in qt(0.975, df = n - 1): NaNs produced

jans <- bs.data[grep("jan", bs.data$EBM.ref, ignore.case = TRUE),]  
jans.VDs <- MASSunite(HFC = jans$HCFC, BM = jans$BM.g, eqn = "quadruped", pch = 19, col = brewer.pal(length(meth.pch), name = "Dark2")[1])  
colbs.quad <- bs.data[grepl("colb", bs.data$EBM.ref, ignore.case = TRUE) & bs.data$Gait == "quad",]  
colbs.quad.VDs <- MASSunite(HFC = colbs.quad$HCFC, BM = colbs.quad$BM.g, eqn = "quadruped", pch = 15, col = brewer.pal(length(meth.pch), name = "Dark2")[1])

## Warning in MASSunite(HFC = colbs.quad$HCFC, BM = colbs.quad$BM.g, eqn =  
## "quadruped", : One or more of the models are inconsistent with ES expectations

colbs.bip <- bs.data[grepl("colb", bs.data$EBM.ref, ignore.case = TRUE) & bs.data$Gait == "bip",]  
colbs.bip.VDs <- MASSunite(FC = colbs.bip$FC, BM = colbs.bip$BM.g, eqn = "biped", pch = 17, col = brewer.pal(length(meth.pch), name = "Dark2")[1])  
dev.off()

## png   
## 2

### Minimum Convex Hulling

pdf("Figure 15.pdf", width = 7, height = 3.75)  
mch.quads <- bs.data[which(bs.data$Method == "MCH" & bs.data$Gait == "quad"),]  
mch.quad.VDs <- MASSunite(HFC = mch.quads$HCFC, BM = mch.quads$BM.g, eqn = "quadruped", pch = 19, col = brewer.pal(length(meth.pch), name = "Dark2")[5], line = TRUE, names = mch.quads$Taxon)

## Warning in MASSunite(HFC = mch.quads$HCFC, BM = mch.quads$BM.g, eqn =  
## "quadruped", : One or more of the models are inconsistent with ES expectations

mch.bips <- bs.data[which(bs.data$Method == "MCH" & bs.data$Gait == "bip"),]  
mch.bips.VDs <- MASSunite(FC = mch.bips$FC, BM = mch.bips$BM.g, eqn = "biped", pch = 17, col = brewer.pal(length(meth.pch), name = "Dark2")[5], line = TRUE, names = mch.bips$Taxon)  
dev.off()

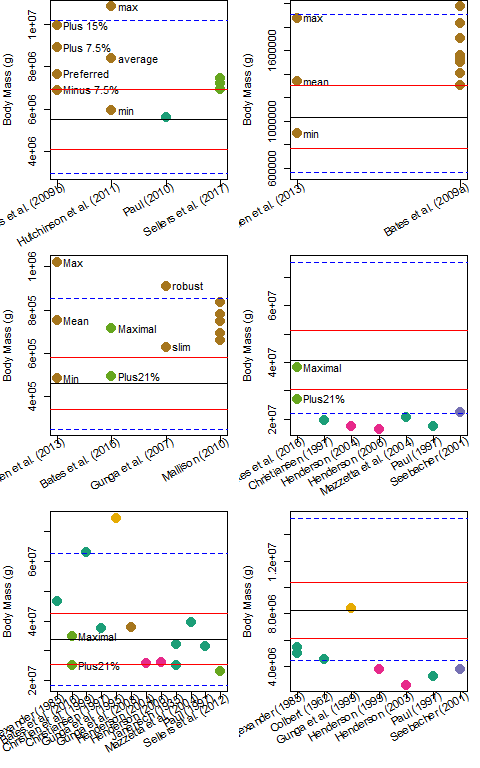
## png   
## 2

# Extra figures

The following pertains to additional analyses and explorations of the data, including specimen-level comparison plots provided in Fig. 9 and the linear vs. quadratic results summarized in Tables 6 and 7. The full list of estimates are generated here, based on the data in Benson *et al.* (2018), compared to the “top 10s” provided in the manuscript tables.

## Specimen-level comparisons

#pdf("Results/case\_examples.pdf", width = 7, height = 9)  
par(mfrow = c(3, 2), mar = c(6,4,0,1))  
  
method.col <- brewer.pal(length(method.order), "Dark2"); names(method.col) <- method.order  
  
specimen.comp <- function(data) {  
 x <- as.factor(data$EBM.ref)  
 masses <- cbind(data[,c("BM.g", "QE", "QE+25%", "QE-25%")], 10^(data$logQE+0.2659), 10^(data$logQE-0.2659))  
 plot(as.numeric(x), data$BM.g, pch = 19, xaxt = "n", xlab = "", ylim = range(masses), col = method.col[data$Method], cex = 2, ylab = "Body Mass (g)")  
 text(as.numeric(as.factor(data$EBM.ref)), data$BM.g, labels = data$BM.Qual, pos = 4)  
 axis(1, at = 1:length(levels(x)), labels = FALSE)  
 text(1:length(levels(x)), par("usr")[3]\*0.9, labels = levels(x), pos = 2, offset = -0.5, srt = 30, xpd = TRUE)  
 abline(h = masses[1,2:6], col = c("black", "red", "red", "blue", "blue"), lty = c(1,1,1,2,2))  
}  
  
  
  
specimen.comp(bs.data[which(bs.data$Meas.sp == "BHI 3033"),]) #tyrannosaurus  
specimen.comp(bs.data[which(bs.data$Meas.sp == "MOR 693"),]) #Allosaurus  
specimen.comp(bs.data[which(bs.data$Meas.sp == "AMNH 2107"),]) #Plateosaurus  
specimen.comp(bs.data[which(bs.data$Taxon == "Apatosaurus louisae"),]) #Apatosaurus  
specimen.comp(bs.data[which(bs.data$Taxon == "Giraffatitan brancai"),]) #Brachiosaurus  
specimen.comp(bs.data[which(bs.data$Taxon == "Iguanodon bernisartensis"),]) #Iguanodon



#dev.off()  
  
#others explored (not run here)  
#specimen.comp(bs.data[which(bs.data$Meas.sp == "CM 9380"),]) #CM 9380  
#specimen.comp(bs.data[which(bs.data$Meas.sp == "BMR P2002.4.1"),]) #CM  
#specimen.comp(bs.data[which(bs.data$Meas.sp == "FMNH PR 2081"),]) #CM  
#specimen.comp(bs.data[which(bs.data$Meas.sp == "MOR 555"),]) #CM  
#specimen.comp(bs.data[which(bs.data$Taxon == "Diplodocus carnegiei"),]) #Diplodocus  
#specimen.comp(bs.data[which(bs.data$Taxon == "Coelophysis bauri"),]) #Coelophysis  
#specimen.comp(bs.data[which(bs.data$Taxon == "Plateosaurus engelhardti"),]) #Acrocanthosaurus

## QE vs. qQE comparisons

# using Benson et al. 2018  
QEqQE.benson <- as.data.frame(matrix(nrow = nrow(benson.data), ncol = 7, dimnames = list(NULL, c("Clade", "Subclade", "Taxon", "Sp", "Juvenile", "FC", "HC"))))  
for(i in 1:nrow(benson.data)) {  
 if(benson.data$Clade[i] == "Theropoda") QEqQE.benson[i, "Gait"] <- "bip"  
 else QEqQE.benson[i, "Gait"] <- benson.data$Gait[i]  
 if(is.na(benson.data$FC[i])) QEqQE.benson[i,"FC"] <- benson.data$FCe[i]  
 else QEqQE.benson[i,"FC"] <- benson.data$FC[i]  
 if(is.na(benson.data$HC[i])) QEqQE.benson[i,"HC"] <- benson.data$HCe[i]  
 else QEqQE.benson[i,"HC"] <- benson.data$HC[i]  
 QEqQE.benson[i,1:5] <- benson.data[i,c("Clade", "Subclade", "Taxon", "Specimen", "Juvenile")]  
}  
  
  
benson.quad <- QE(HC = QEqQE.benson$HC[which(QEqQE.benson$Gait == "quad")], FC = QEqQE.benson$FC[which(QEqQE.benson$Gait == "quad")], equation = "raw", quadratic = TRUE, data = QEqQE.benson[which(QEqQE.benson$Gait == "quad"),])  
#sorted by size  
benson.quad <- benson.quad[order(benson.quad$QE, decreasing = TRUE),] #toggle TRUE and FALSE to sort data by increase or decreasing size  
  
benson.bip <- cQE(FC = QEqQE.benson$FC[which(QEqQE.benson$Gait == "bip")],equation = "raw", quadratic = TRUE, data = QEqQE.benson[which(QEqQE.benson$Gait == "bip"),])  
benson.bip <- benson.bip[order(benson.bip$cQE, decreasing = FALSE),] #toggle TRUE and FALSE to sort data by increase or decreasing size