

# Current and potential roles for archaeology in the development of cultural evolutionary theory (published in Philosophical Transactions B)

Raven Garvey

17 November, 2017

## Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Multi panel Washita figure</b>	<b>2</b>
2.1	Create custom colors . . . . .	2
2.2	Plot and simulation parameters . . . . .	2
2.3	The plot . . . . .	3
2.4	Kernel density plot to compare simulation results . . . . .	9

## 1 Introduction

Code for research published in:

Garvey Raven Current and potential roles of archaeology in the development of cultural evolutionary theory. *Phil. Trans. R. Soc. B ans. R. Soc. B* <http://doi.org/10.1098/rstb.2017.0057>

These electronic supplementary materials are also available online at <https://dx.doi.org/10.6084/m9.figshare.c.3965853>.

```
data <- read.csv("data/data.csv", skip = 6)
str(data)
```

```
## 'data.frame':   259 obs. of  9 variables:
## $ wash_weight      : num  0.29 0.29 0.3 0.3 0.31 0.32 0.4 0.4 0.4 0.4 ...
## $ wash_thick       : num  1.97 2.05 2.1 2.18 2.24 2.27 2.28 2.28 2.29 2.29 ...
## $ wash_length      : num  12.6 13.9 14.1 14.1 14.3 ...
## $ wash_midlength   : num  12.3 13 13.1 13.2 13.2 ...
## $ wash_width       : num  8.55 8.81 9.19 9.24 9.47 ...
## $ wash_basewidth   : num  7.71 7.84 8.17 8.51 8.76 8.8 8.87 8.88 8.99 9 ...
## $ wash_bladelength : num  6.6 7.01 8.28 8.66 8.83 8.88 8.98 9.38 9.5 9.68 ...
## $ wash_haftlength  : num  4.02 4.1 4.23 4.25 4.3 4.41 4.79 4.83 4.86 4.87 ...
## $ wash_neck        : num  4.01 4.15 4.3 4.35 4.36 4.36 4.39 4.46 4.48 4.55 ...
```

## 2 Multi panel Washita figure

### 2.1 Create custom colors

```
makeTransparent <- function(black, alpha = 200){  
  newColor <- col2rgb(black)  
  apply(newColor, 2, function(curcoldata)  
    {rgb(red = curcoldata[1],  
        green = curcoldata[2],  
        blue = curcoldata[3],  
        alpha = alpha, maxColorValue = 255)})  
}  
tBlack <- makeTransparent("gray40")  
tOrange <- makeTransparent("orange")  
tBlue <- makeTransparent("cornflowerblue")
```

### 2.2 Plot and simulation parameters

Make data frame

```
D <- data.frame(data$wash_width, data$wash_basewidth,  
               data$wash_neck, data$wash_haftlength,  
               data$wash_length, data$wash_midlength,  
               data$wash_thick, data$wash_bladelength,  
               data$wash_weight)
```

Set counter to # variables (i.e., # figure rows)

```
count <- c(1:9)
```

Set the attribute-specific n.measures

```
n.measures <- c(162, 157,  
                249, 227,  
                136, 136,  
                259, 135,  
                87)
```

y-axis panel plot labels

```
labs <- c("width\n(mm)", "base width\n(mm)",  
          "neck width\n(mm)", "haft length\n(mm)",  
          "length\n(mm)", "mid length\n(mm)",  
          "thickness\n(mm)", "blade length\n(mm)",  
          "weight\n(g)")
```

Simulation parameters

```

set.seed(1212)          # set seed
n.houses      <- 100    # set number of houses at site
y             <- 1000   # set number of simulation runs

## plotting parameters
col.density.1 <- tBlack  # set density color for 10%
col.outline.1 <- tBlack  # set density outline color 10%
col.density.05 <- tOrange # set density color for 5%
col.outline.05 <- tOrange # set density outline color 5%
col.density.03 <- tBlue  # set density color for 3%
col.outline.03 <- tBlue  # set density outline color 3%
col.mean       <- "black" # set col for mean of arch sample
cex.ylab       <- 1.2     # set y-axis label size
cex.ylabtick   <- 1.1     # set y-axis tick label size
cex.xlabtick   <- 1.1     # set x-axis tick label size

x.lim2 <- c(6, 7,        # set x limit for each plot
            3.5, 4,      # (this also affects tick mark placement)
            50, 40,
            1.4, 35,
            0.12)

y.lim2 <- c(1.2, 1,      # set y limit for each plot
            1.75, 1.2,
            0.15, 0.15,
            5, 0.15,
            45)

```

## 2.3 The plot

```

pdf(file="figures/Garvey_Figure3a.pdf",
    height = 12, width = 12)

par(mfcol = c(length(count), 3))
par(oma = c(6, 10, 5, 2))

# column 1 (SD = 0.1)

diff10 <- NULL #create empty vector to hold calculate diff between
              #simulated population mean and observed mean (standardized)

for(x in count) { # looping procedure for each plotting pannel

SD      <- 0.1    # set standard deviation multiplier for random draws
d       <- na.omit(D[, x]) # select the data set to sample
N       <- n.measures[x]   # select the appropriate n.measures

metadig <- NULL #create empty vector to hold digs
var.Kdigs <- NULL #create empty vector to hold variances
cv.Kdigs <- NULL #create empty vector to hold CVs

```

```

for(i in 1:y) {      # "master" loop that repeats
                    # "archaeological sampling" from
                    # "generational learning sample" one thousand times

    all.houses <- NULL      # create an empty vector to hold results

    for(j in 1:n.houses) {      #loop through all houses

        house.seed <- sample(d, 1, replace = TRUE)
        house.g1 <- rnorm(4, mean = house.seed,
                          sd = SD * mean(house.seed))
        house.g2 <- rnorm(4, mean = mean(house.g1),
                          sd = SD * mean(house.g1))
        house.g3 <- rnorm(4, mean = mean(house.g2),
                          sd = SD * mean(house.g2))
        house.g4 <- rnorm(4, mean = mean(house.g3),
                          sd = SD * mean(house.g3))
        total.house.sample <- as.vector(c(house.g1,
                                           house.g2,
                                           house.g3,
                                           house.g4))
        all.houses <- c(total.house.sample, all.houses)

    }

    dig <- sample(all.houses, N, replace = TRUE)
    metadig <- c(metadig, dig)
    var.Kdigs <- c(var(dig), var.Kdigs)
    cv.Kdigs <- c(sd(dig)/mean(dig), cv.Kdigs)
}

diff10 <- c(diff10, ((var(d)-mean(var.Kdigs))/sd(var.Kdigs)))

#plotting
dp.md <- density(var.Kdigs)
par(mai = c(0.25, 0.5, 0.1, 0))

plot(dp.md, col = col.outline.1, bty = 'n',
     main = "", xlab = "", ylab = "",
     xlim = c(0, x.lim2[x]),
     ylim = c(0, y.lim2[x]),
     xaxt = "n", yaxt = "n")

axis(1, at = c(0, 0.25 * x.lim2[x],
               0.5 * x.lim2[x], 0.75 * x.lim2[x],
               x.lim2[x]),
     cex = cex.xlabtick)

# alternative, with only 3 x-axis ticks
# axis(1, at = c(0, 0.5 * x.lim2[x],
#               x.lim2[x]),
#       cex = cex.xlabtick)

```

```

axis(2, at = c(0, 0.5 * y.lim2[x],
               y.lim2[x]),
     line = 1.5, las = 2,
     cex = cex.ylabtick)

polygon(dp.md, col = col.density.1)
abline(v = var(d), col = col.mean, lwd = 2)
mtext (text = labs[x], side = 2,
       las = 1, line = 5.5,
       cex = 1.2, outer = FALSE)

}

mtext (text = "A. CV = 10%", side = 3,
       las = 1, line = 73,
       cex = 1.5, outer = FALSE)

# column 2 (CV = 5%)

diff5      <- NULL  #create empty vector to hold calculate diff between
                      #simulated population mean and observed mean (standardized)

for(x in count) {  # looping procedure for each plotting pannel

SD          <- 0.05  # set standard deviation multiplier for random draws
d           <- na.omit(D[, x])  # select the data set to sample
N           <- n.measures[x]    # select the appropriate n.measures

metadig     <- NULL  #create empty vector to hold digs
var.Kdigs   <- NULL  #create empty vector to hold variances
cv.Kdigs    <- NULL  #create empty vector to hold CVs

for(i in 1:y) {    # "master" loop that should repeat
                      # "archaeological sampling" from
                      # "generational learning sample" one thousand times

  all.houses <- NULL  # create an empty vector to hold results

  for(j in 1:n.houses) {  #loop through all houses

    house.seed <- sample(d, 1, replace = TRUE)
    house.g1 <- rnorm(4, mean = house.seed,
                     sd = SD * mean(house.seed))
    house.g2 <- rnorm(4, mean = mean(house.g1),
                     sd = SD * mean(house.g1))
    house.g3 <- rnorm(4, mean = mean(house.g2),
                     sd = SD * mean(house.g2))
    house.g4 <- rnorm(4, mean = mean(house.g3),
                     sd = SD * mean(house.g3))
    total.house.sample <- as.vector(c(house.g1,
                                      house.g2,

```

```

                                house.g3,
                                house.g4))
all.houses <- c(total.house.sample, all.houses)

}
dig <- sample(all.houses, N, replace = TRUE)
metadig <- c(metadig, dig)
var.Kdigs <- c(var(dig), var.Kdigs)
cv.Kdigs <- c(sd(dig)/mean(dig), cv.Kdigs)
}

diff5 <- c(diff5, ((var(d)-mean(var.Kdigs))/sd(var.Kdigs)))

#plotting
dp.md <- density(var.Kdigs)
par(mai = c(0.25, 0.5, 0.1, 0))

plot(dp.md, col = col.outline.05, bty = 'n',
     main = "", xlab = "", ylab = "",
     xlim = c (0, x.lim2[x]),
     ylim = c (0, y.lim2[x]),
     xaxt = "n", yaxt = "n")

axis(1, at = c(0, 0.25 * x.lim2[x],
               0.5 * x.lim2[x], 0.75 * x.lim2[x],
               x.lim2[x]),
     cex = cex.xlabtick)

polygon(dp.md, col = col.density.05)
abline(v = var(d), col = col.mean, lwd = 2)

}

mtext (text = "B. CV = 5%", side = 3,
      las = 1, line = 73,
      cex = 1.5, outer = FALSE)

# column 3 (CV = 3%)

diff3      <- NULL  #create empty vector to hold calculate diff between
                    #simulated population mean and observed mean (standardized)

for(x in count) {  # looping procedure for each plotting pannel

SD      <- 0.03    # set standard deviation multiplier for random draws
d       <- na.omit(D[, x])  # select the data set to sample
N       <- n.measures[x]    # select the appropriate n.measures

```

```

metadig <- NULL #create empty vector to hold digs
var.Kdigs <- NULL #create empty vector to hold variances
cv.Kdigs <- NULL #create empty vector to hold CVs

for(i in 1:y) { # "master" loop that should repeat
  # "archaeological sampling" from
  # "generational learning sample" one thousand times

  all.houses <- NULL # create an empty vector to hold results

  for(j in 1:n.houses) { #loop through all houses

    house.seed <- sample(d, 1, replace = TRUE)
    house.g1 <- rnorm(4, mean = house.seed,
                     sd = SD * mean(house.seed))
    house.g2 <- rnorm(4, mean = mean(house.g1),
                     sd = SD * mean(house.g1))
    house.g3 <- rnorm(4, mean = mean(house.g2),
                     sd = SD * mean(house.g2))
    house.g4 <- rnorm(4, mean = mean(house.g3),
                     sd = SD * mean(house.g3))
    total.house.sample <- as.vector(c(house.g1,
                                     house.g2,
                                     house.g3,
                                     house.g4))
    all.houses <- c(total.house.sample, all.houses)

  }

  dig <- sample(all.houses, N, replace = TRUE)
  metadig <- c(metadig, dig)
  var.Kdigs <- c(var(dig), var.Kdigs)
  cv.Kdigs <- c(sd(dig)/mean(dig), cv.Kdigs)
}

diff3 <- c(diff3, ((var(d)-mean(var.Kdigs))/sd(var.Kdigs)))

#plotting
dp.md <- density(var.Kdigs)
par(mai = c(0.25, 0.5, 0.1, 0))

plot(dp.md, col = col.outline.03, bty = 'n',
     main = "", xlab = "", ylab = "",
     xlim = c(0, x.lim2[x]),
     ylim = c(0, y.lim2[x]),
     xaxt = "n", yaxt = "n")

axis(1, at = c(0, 0.25 * x.lim2[x],
               0.5 * x.lim2[x], 0.75 * x.lim2[x],
               x.lim2[x]),
     cex = cex.xlabtick)

```

```
polygon(dp.md, col = col.density.03)
abline(v = var(d), col = col.mean, lwd = 2)

}
```

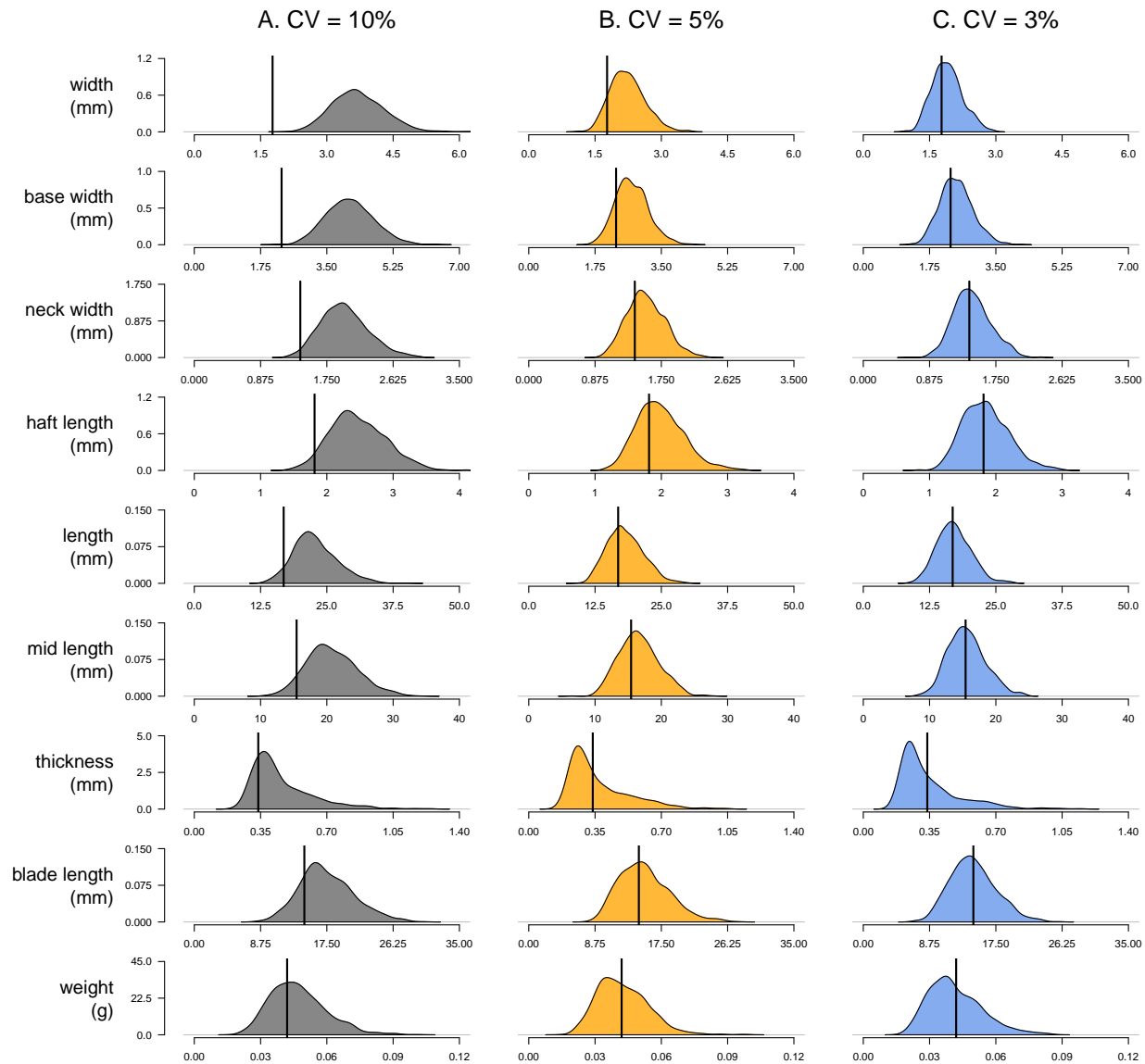
```
mtext (text = "C. CV = 3%", side = 3,
       las = 1, line = 73,
       cex = 1.5, outer = FALSE)
```

```
dev.off()
```

```
## pdf
## 2
```

```
knitr::include_graphics("figures/Garvey_Figure3a.pdf")
```





## 2.4 Kernel density plot to compare simulation results

```
pdf(file="figures/Garvey_Figure3b.pdf",
    height = 3, width = 6)

d10 <- density(diff10)
d5 <- density(diff5)
d3 <- density(diff3)

plot(d10, xlim = c(-5, 1),
     ylim = c(0, 4),
     xlab = "",
```

```

    main = "", las = 1)
polygon(d10, col = tBlack)
lines(d5)
polygon(d5, col = tOrange)
lines(d3)
polygon(d3, col = tBlue)
mtext(side = 1,
      "standardized distance from archaeological sample\n (Z-score)",
      outer = FALSE,
      line = 3.5)

legend("topleft",
      legend = c("CV = 10%",
                  "CV = 5%",
                  "CV = 3%"),
      fill = c(tBlack, tOrange, tBlue),
      bty = "n" )

dev.off()

```

```

## pdf
## 2

```

```

knitr::include_graphics("figures/Garvey_Figure3b.pdf")

```

