End Member Mixing Analysis

Performed with EMMAgeo in R

Author: Andrew Macumber

Table of Contents

Prep Workspace

Remove previous lists from R

```
rm(list=ls())
```

Load libraries

```
library("EMMAgeo")
```

```
##
## EMMAgeo v. 0.9.6. When using this package please cite:
##
## Dietze, E., and Dietze, M.: Grain-size distribution unmixing using the R package EMMAgeo, E&G Quatern
##
## Dietze E, Hartmann K, Diekmann B, IJmker J, Lehmkuhl F, Opitz S, Stauch G, Wuennemann B, Borchers A.
##
## (Don't forget to mention the package and R version.)
```

```
library("functional")
```

Load Data

Bring in the dataset

```r
psa_v000 <- read.table('HGC2_Shallow_PSA.csv', sep = ",")
```

Check dataframe (optional)

```r
dim(psa_v000)
head(psa_v000)
```

Data Wrangling

Remove one row and two column headers

```r
psa_v001 <- psa_v000[-1, -c(1,2)]
```

Check for grain size bins (columns) with zero sum

```r
colSums(psa_v001)
```

```
##         V3         V4         V5         V6         V7         V8 
##   6.397511  11.389342  16.797859  23.975903  29.974757  35.318425 
##         V9        V10        V11        V12        V13        V14 
##  40.267415  45.175638  49.479315  53.154473  56.391264  59.404969 
##        V15        V16        V17        V18        V19        V20 
##  62.403159  65.275558  68.240213  71.430199  75.224878  79.583271 
##        V21        V22        V23        V24        V25        V26 
##  84.623755  90.382430  97.018026 104.529371 112.915502 122.175530 
##        V27        V28        V29        V30        V31        V32 
## 132.234464 143.039421 154.494820 166.643975 179.504054 193.128960 
##        V33        V34        V35        V36        V37        V38 
## 207.502347 222.747497 239.190299 257.253585 277.181638 298.876336 
##        V39        V40        V41        V42        V43        V44 
## 322.129283 346.830570 372.928676 399.638422 424.899085 445.523453 
##        V45        V46        V47        V48        V49        V50 
## 458.555299 462.977032 460.077122 452.761402 443.735666 434.184590 
##        V51        V52        V53        V54        V55        V56 
## 423.923097 411.432282 394.494106 370.226342 334.821571 285.525388 
##        V57        V58        V59        V60        V61        V62 
## 223.266571 154.200687  88.333760  39.634783  12.994591   3.085919 
##        V63        V64        V65 
##   0.460077   0.033709   0.000482
```

Remove NULL values

```r
psa_v002 <- na.omit(psa_v001)
```

Convert to double matrix

```r
psa_v003 <- as.matrix(psa_v002)
```

Ensure rows sum to 100

```
psa_v004 <- psa_v003 / apply(psa_v003,1,sum) * 100
```

Check dataframe (optional)

```
dim(psa_v004)
head(psa_v004)
```

Data Objects

Create a grain size vector

```
mm_bins <- as.numeric(psa_v000[1,-c(1,2)])
```

Create a depth vector

```
depth <- as.vector(psa_v000[-1,1])
```

Assign an emma object

```
emma <- psa_v004
```

Graph: Grain Size Frequency Distribution

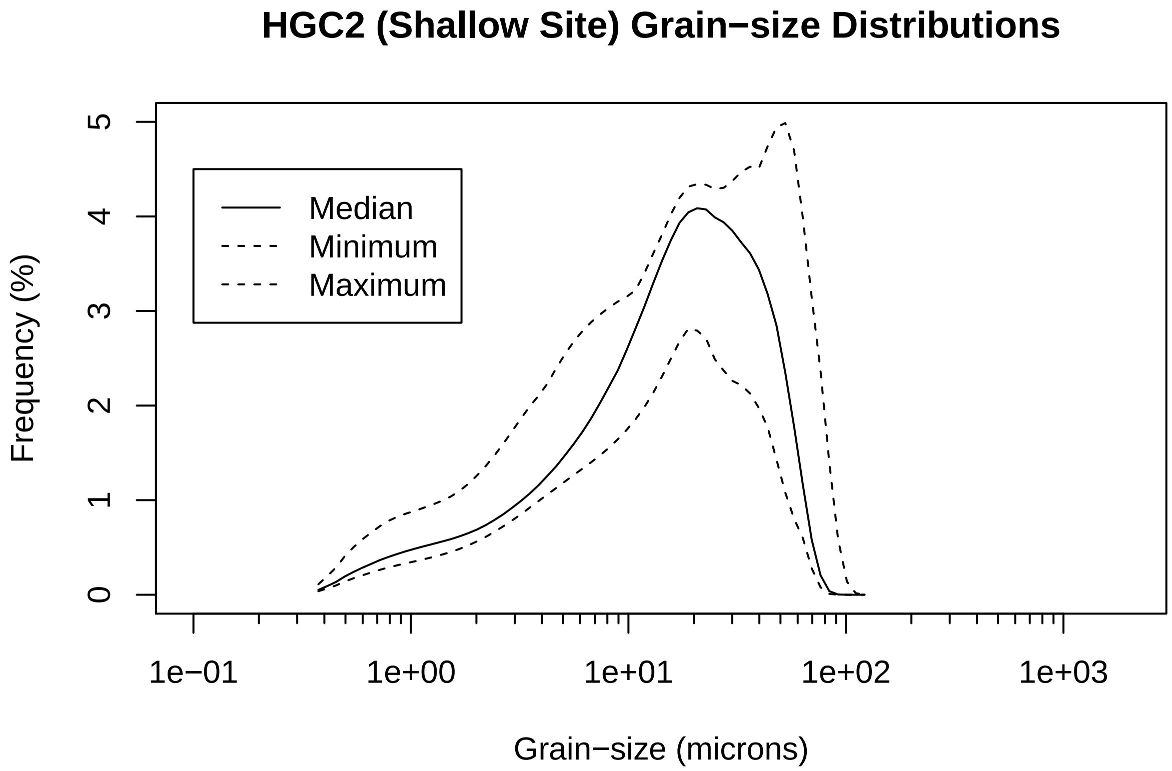
Generate a plot of median, minimum and maximum grain size (log) versus frequency (%)

Create a dataframe with grain size (column) mins, max and medians

```
library("matrixStats")
psa.range <- cbind(mm_bins,
                   colMins(emma, na.rm=TRUE),
                   colMaxs(emma, na.rm=TRUE),
                   colMedians(emma, na.rm=TRUE))
```

Create a graph showing the median, minimum and maximum grain size frequencies

```
plot(psa.range[,1],psa.range[,4], type ="l", log = "x", # plot grain-size (log) versus median
     xlim = c(0.1,2000), ylim = c(0,5), # set the axes limits
     col ="black", xaxt = "n", # colour of the line and suppress x-axis
     xlab = "Grain-size (microns)",
     ylab = "Frequency (%)",
     main = "HGC2 (Shallow Site) Grain-size Distributions") # labels
lines(psa.range[,1],psa.range[,2], col = "black", lty = 2) # add a line for minimum (grey-dash)
lines(psa.range[,1],psa.range[,3], col = "black", lty = 2) # add a line for maximum (grey-dash)
axis (side=1, at = c(0.1,1,10,100,1000)) # add major tick marks
axis (side=1, at = c(seq(0.2,0.9,by = 0.1), seq(2,9, by =1), # add minor tick marks
                     seq(20,90, by = 10), seq(200,900,by=100)),
      labels = NA, tcl=-0.25, lwd=0, lwd.ticks=1) # no labels and shorten tick marks
legend(0.1,4.5,c("Median","Minimum","Maximum"), # add a legend at specific spot with labels
       lty=c(1,2,2), col=c("black","black","black")) # ensure same as lines in plot
```



EMMA: Model Parameters

Check your dataset for errors.

```
check.data(X = emma , q = 10, l = seq(0, 1, 0.01), c = 100)
```

```
## [1] "Data matrix passed test... OK"
## [2] "End-member vector passed test... OK"
## [3] "Weight transformation limit vector passed test... OK"
## [4] "Scaling parameter passed test... OK"
## [5] "NA-test passed... OK"
## [6] "Test for zero-only values passed... OK"
## [7] "    Note: weight transformation limit(s) are out of range. Maximum value is 0.01"
## [8] "All samples sum up to constant sum... OK"
```

Create a sequence of weight transformations (l)

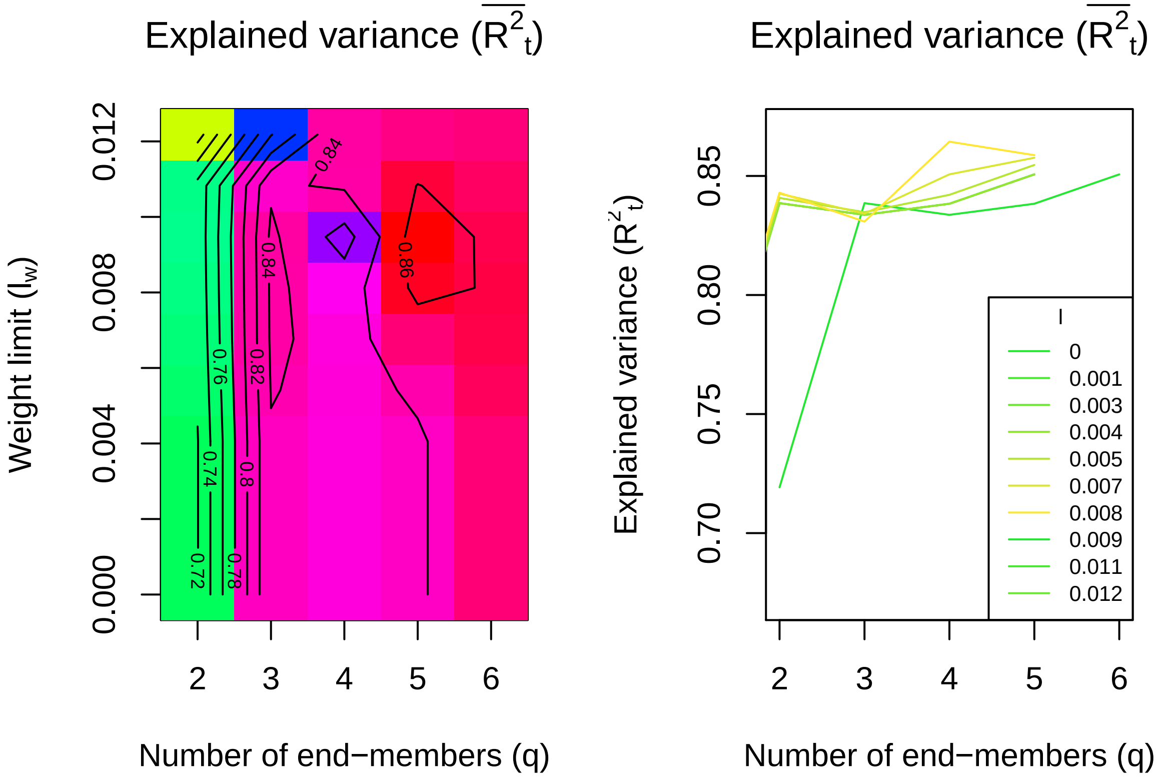
```
l <- get.l(X = emma, n = 10, max = 0.95, min = 0)
```

Combine the range of end members and weight transformations (Q). Create a range of end members only (q)

```
Q <- get.q(X = emma, l = l)
q <- 2:6
```

See a graph showing how weight transformation (l) varies with number of end Members

```
EM.tp <- test.parameters(X = emma, q = q, l = l, plot = "mRt",
                         legend = "bottomright", cex = 0.7,
                         multicore = FALSE,
                         colour = rgb((1:7) / 7, 0.9, 0.2, 1)
                         )
```



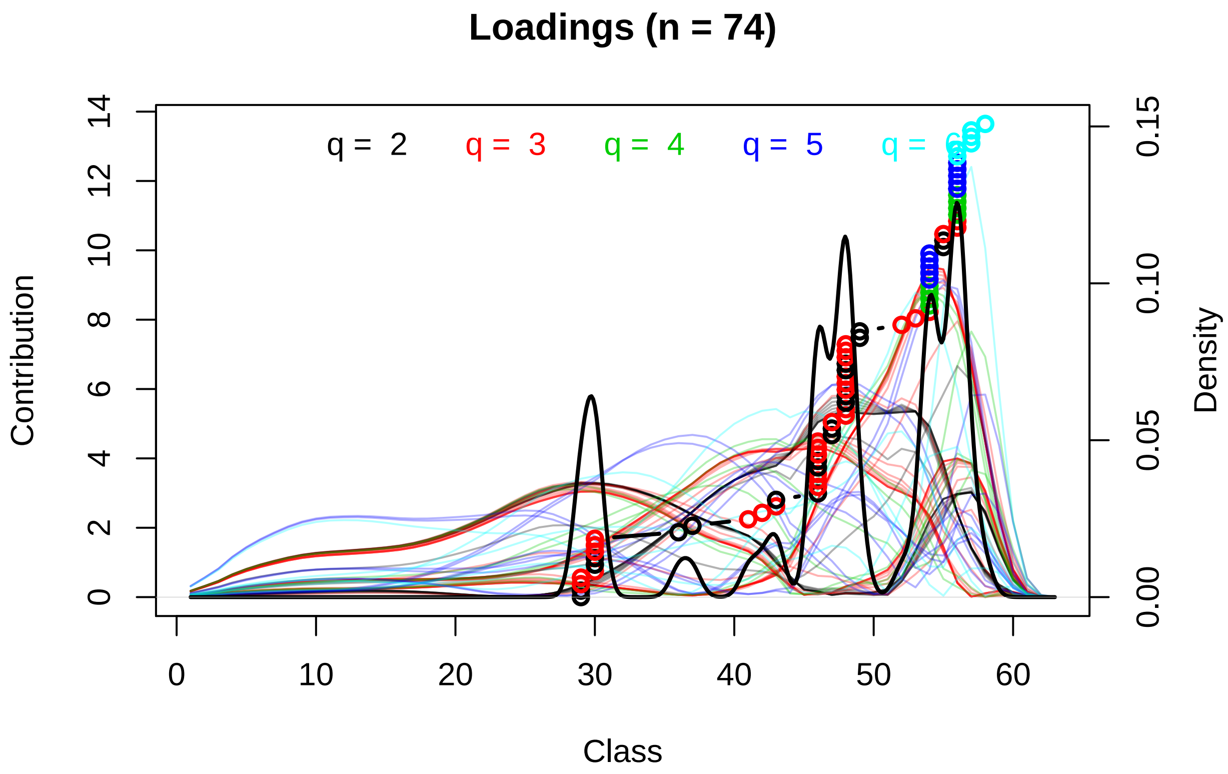
See a table showing how weight transformation (l) varies with number of end-members

```
EM.tp$mRt
```

```
##             0 0.00135327634978331 0.00270655269956662 0.00405982904934993
## 2 0.7192303           0.7192303           0.7192303           0.7192303
## 3 0.8385825           0.8385825           0.8385825           0.8385825
## 4 0.8336792           0.8336792           0.8336792           0.8336792
## 5 0.8383330           0.8383330           0.8383330           0.8383330
## 6 0.8506654           0.8506654           0.8506654           0.8506654
##   0.00541310539913324 0.00676638174891655 0.00811965809869986
## 2           0.7218882           0.7244840           0.7262545
## 3           0.8407663           0.8425840           0.8429593
## 4           0.8347849           0.8341638           0.8308291
## 5           0.8420749           0.8506727           0.8644017
## 6           0.8546084           0.8576504           0.8587276
##   0.00947293444848317 0.0108262107982665 0.0121794871480498
## 2           0.7273227          0.7268903          0.6714122
## 3           0.8435900          0.8372189          0.7788796
## 4           0.8118646          0.8425722          0.8436577
## 5           0.8701486          0.8603779          0.8481795
## 6           0.8568911          0.8538333          0.8502024
```

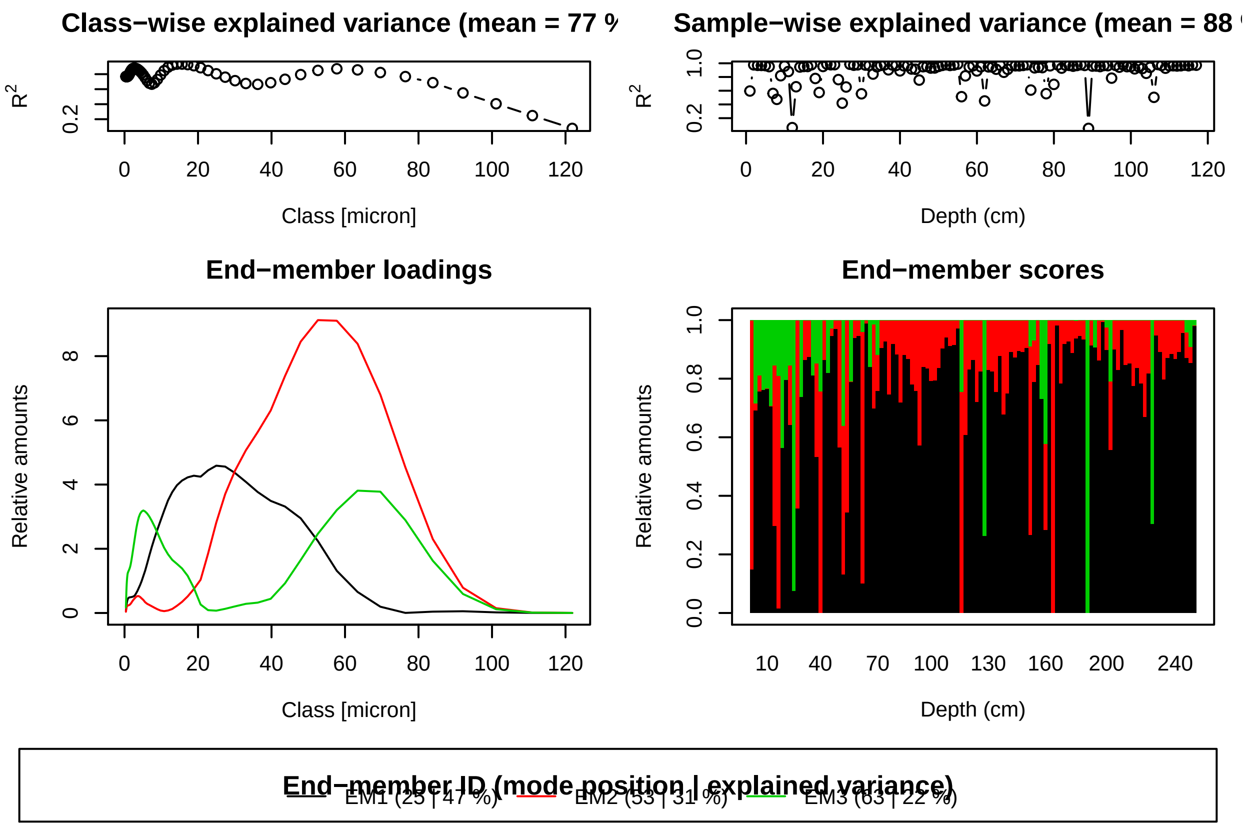
EMMA: Run All Models

```
EM.pot <- model.EM(X = emma, q = Q)
```



EMMA: Run Specific Model

```
EM.opt <- EMMA(X = emma,
               q = 3, l = 0.009,
               #q = 3, l = 0.012,
               c = 100,
               plot = TRUE,
               classunits = mm_bins,
               ID = as.numeric(depth),
               xlab = c(expression(paste("Class [", micron, "]")), "Depth (cm)"))
```



EMMA: Run Robust Model

Get the bin size ranges for end members (i.e., limits)

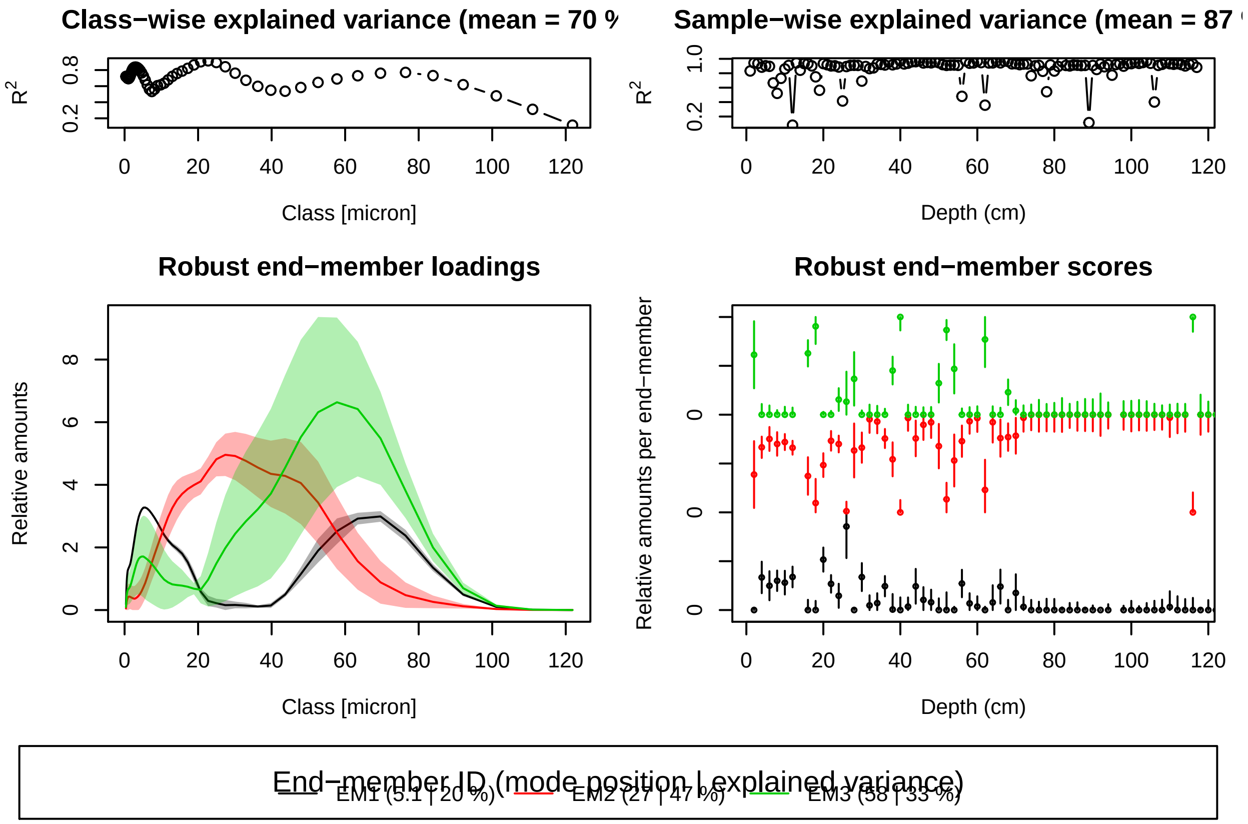
```
# Visualize limits
stem(EM.pot$modes, scale = 2)
```

```
##
##   The decimal point is at the |
##
##   28 | 0000
##   30 | 000000
##   32 |
##   34 |
##   36 | 00
##   38 |
##   40 | 0
##   42 | 000
##   44 |
##   46 | 000000000000
##   48 | 00000000000000
##   50 |
##   52 | 00
##   54 | 0000000000000
##   56 | 0000000000000000
##   58 | 0
```

```r
# Assign limits
v1 <- c(28,46,54)
v2 <- c(30,48,56)
limits = cbind(v1,v2)
```

Run robust model

```r
EM.rob <- robust.EM(em = EM.pot
                    , l = 0.009
                    , limits = limits
                    , c = 100
                    , plot = TRUE
                    , classunits = mm_bins
                    , ID = as.numeric(depth)
                    , xlab = c(expression(paste("Class [", micron, "]")), "Depth (cm)")
                    )
```



EMMA: Results

We tested 74 models to find the initial parameters. The mean total explained variance:

- For all models: 0.8603779
- For the optimal model: 0.8267733
- For the robust model: 0.7862913

The optimal model had the following end member modes: 24.9538, 52.6264, 63.4192 each explaining 47.0805236, 30.6214553, 22.2980211 of the variance, respectively.

The robust model had the following end member modes: 5.11087, 27.3934, and 57.7713. Each explaining 20.4586628, 46.8260779, 32.7152593 of the variance, respectively.

Save Data

Create a dataframe with robust end member mean abundance and standard deviation.

```
EM.rob.mean <- EM.rob$scores$mean
EM.rob.sd <- EM.rob$scores$sd
EM.rob.scores <- cbind(EM.rob.mean, EM.rob.sd)
colnames(EM.rob.scores) <- c("EM_05_Mean", "EM_27_Mean", "EM_58_Mean",
                             "EM_05_SD", "EM_27_SD", "EM_58_SD")
row.names(EM.rob.scores) <- depth
```

Export File

```
write.csv(EM.rob.scores, file ="HGC2_EMMA_Q3_scores.csv")
```

Graph: Stacked Stratigraphic Plot

Create a stacked stratigraphic plot of robust end member abundance. You will need to format a dataset using "HGC2_EMMA_Q3_scores.csv" that confirms to three columns T, Var, Val: * T - is depth or time * Var - is the variable, that is the end member * Val - is the value or abundance of that end member at that depth or time

For example:/ T VAR VAL
0 EM01 0.25
0 EM02 0.50
0 EM03 0.25
1 EM01 0.10
2 EM02 0.85
3 EM03 0.05

Load Libraries & Data

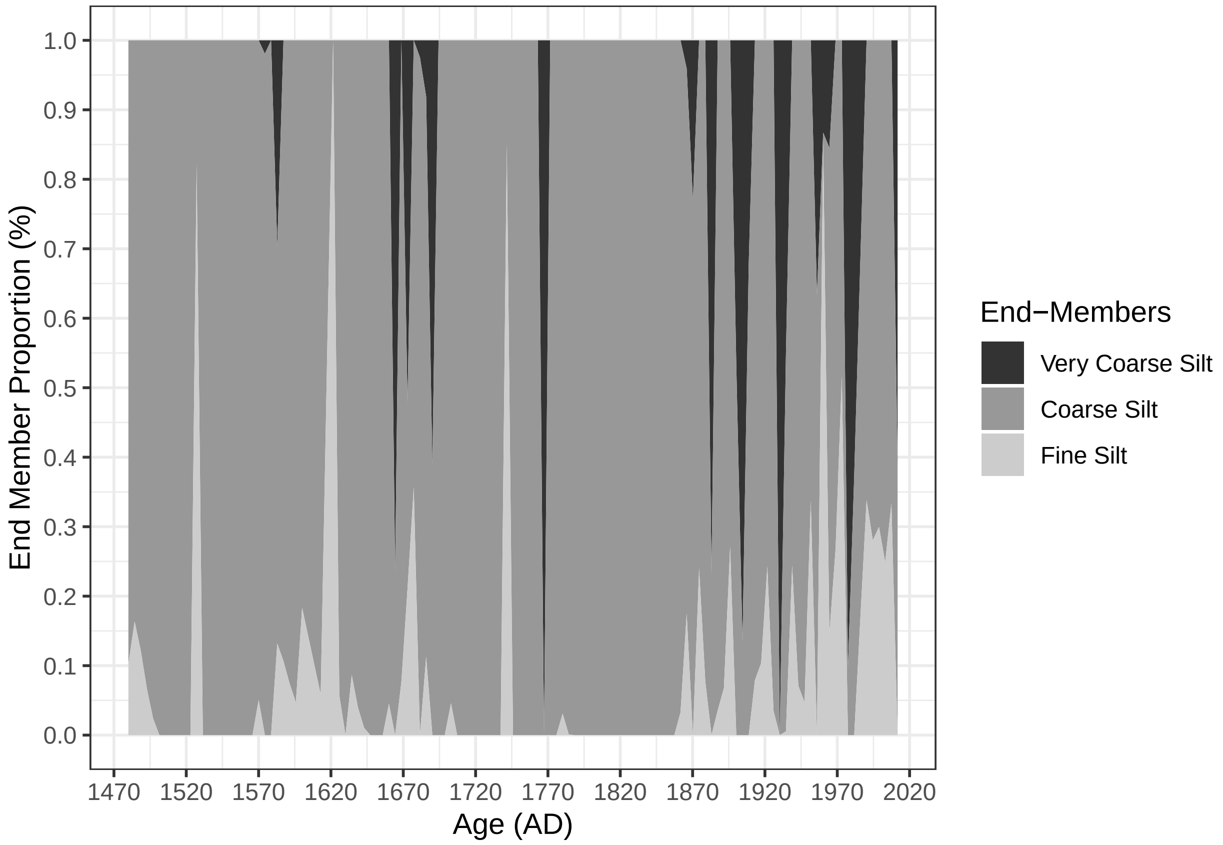
```
library(ggplot2)
d <- read.table('HGC2_EMMA_Q3_StackedInput_Age.csv', sep = ",", header = TRUE)
```

Create the plot: 1. Create the area plot 2. Change breaks along the x-axis 3. Change the axes labels 4. Change legend labels and make grayscale

```
sa1 <- ggplot(d,aes(x=T,y=VAL,group=VAR,fill=VAR)) + geom_area(position="fill")
sa2 <- sa1 + scale_x_continuous(breaks=seq(1420,2020,50)) + scale_y_continuous(breaks=seq(0,1,1/10))
sa3 <- sa2 + labs(x="Age (AD)",
                  y="End Member Proportion (%)")
sa4 <- sa3 + scale_fill_grey(name = "End-Members", labels = c("Very Coarse Silt", "Coarse Silt", "Fine 
```

View Plot

```
sa4
```



Graph: Robust End Member Distributions

This figure will show the average grain size distribution with standard deviations for each robust end member. Each end member will be plotted on its on graph. The grain size distributions of all samples will be displayed as grey lines on each graph. *Note: Creates Figure 3 of Chapter 2. Markdown creates an improperly formatted graph. Not run here*

Format sample data

```
dietze <- rbind(mm_bins, emma)
dietze <- t(dietze)
```

Format robust end member means

```
robust <- rbind(mm_bins, EM.rob$loadings$mean)
robust <- t(robust)
```

Format robust end member standard deviations

```
robust.sd <- t(EM.rob$loadings$sd)
```

Plot the graph

```
plot.new()

# Design the layout of the graph
layout(matrix(c(1,1,2,2,3,3), 3, 2, byrow = TRUE),
       widths=c(3,3,3), heights=c(2,2,2))

# First Graph

## Samples
plot(dietze[,1],dietze[,2], type = "l", log ="x", ylim = c(0,10)
     , xlim = c(0.1,2000),col="grey"
     , xaxt="n", xlab = ""
     , ylab = "Frequency (%)")
for (i in 3:ncol(dietze)) lines(dietze[,1],dietze[,i],col="grey") # plot for every column

## The End Member
lines(robust[,1],robust[,2],type="l",col="black",lty = 1, lwd = 2) # mean
lines(robust[,1],robust[,2]-robust.sd[,1],type="l",col="black",lty = 2, lwd = 2) # lower standard devia
lines(robust[,1],robust[,2]+robust.sd[,1],type="l",col="black",lty = 2, lwd = 2)

## Graph Details
axis (side=1, at = c(0.1,1,10,100,1000), labels = NA) # add major tick marks
axis (side=1, at = c(seq(0.2,0.9,by = 0.1), seq(2,9, by =1), # add minor tick marks
                     seq(20,90, by = 10), seq(200,900,by=100))
      , labels = NA, tcl=-0.25, lwd=0, lwd.ticks=1) # no labels and shorten tick marks
legend(200,9,c("Samples","End Member Mean","End Member St. Deviation"), # add a legend at specific spot
       lty=c(1,1,2), col=c("grey","black","black")) # ensure same as lines in plot
text(0.1, 8, adj = 0, "Fine Silt End Member (5 microns) Explaining 21% of Variance", cex = 1.5)
title(main = "Robust End Members of Harvey Lake Shallow Core (HGC2)", cex.main = 2)


# Second Graph

## Samples
plot(dietze[,1],dietze[,2], type = "l", log ="x", ylim = c(0,10),
     xlim = c(0.1,2000),col="grey", xaxt="n", xlab = "",
     ylab = "Frequency (%)")
for (i in 3:ncol(dietze)) lines(dietze[,1],dietze[,i],col="grey") # plot for every column

## The End Member
lines(robust[,1],robust[,3],type="l",col="black", lty = 1, lwd = 2)
lines(robust[,1],robust[,3]-robust.sd[,2],type="l",col="black", lty = 2, lwd = 2) # upper standard devi
lines(robust[,1],robust[,3]+robust.sd[,2],type="l",col="black", lty = 2, lwd = 2)

## Graph Details
axis (side=1, at = c(0.1,1,10,100,1000), labels = NA) # add major tick marks
axis (side=1, at = c(seq(0.2,0.9,by = 0.1), seq(2,9, by =1), # add minor tick marks
                     seq(20,90, by = 10), seq(200,900,by=100))
      , labels = NA, tcl=-0.25, lwd=0, lwd.ticks=1) # no labels and shorten tick marks
```

```
legend(200,9,c("Samples","End Member Mean","End Member St. Deviation"), # add a legend at specific spot
       lty=c(1,1,2), col=c("grey","black","black")) # ensure same as lines in plot
text(0.1, 8, adj = 0, "Coarse Silt End Member (27 microns) Explaining 42% of Variance", cex = 1.5)


# Third Graph

## Samples
plot(dietze[,1],dietze[,2], type = "l", log ="x", ylim = c(0,10),
     xlim = c(0.1,2000),col="grey", xaxt="n", xlab = "",
     ylab = "Frequency (%)")
for (i in 3:ncol(dietze)) lines(dietze[,1],dietze[,i],col="grey") # plot for every column

## The End Member
lines(robust[,1],robust[,4],type="l",col="black",lty = 1, lwd = 2)
lines(robust[,1],robust[,4]-robust.sd[,3],type="l",col="black",lty = 2, lwd = 2)
lines(robust[,1],robust[,4]+robust.sd[,3],type="l",col="black",lty = 2, lwd = 2)

## Graph Details
axis (side=1, at = c(0.1,1,10,100,1000)) # add major tick marks
axis (side=1, at = c(seq(0.2,0.9,by = 0.1), seq(2,9, by =1), # add minor tick marks
                     seq(20,90, by = 10), seq(200,900,by=100))
      , labels = NA, tcl=-0.25, lwd=0, lwd.ticks=1) # no labels and shorten tick marks
legend(200,9,c("Samples","End Member Mean","End Member St. Deviation"), # add a legend at specific spot
       lty=c(1,1,2), col=c("grey","black","black")) # ensure same as lines in plot
text(0.1, 8, adj = 0, "Very Coarse Silt End Member (58 microns) Explaining 33% of Variance", cex = 1.5)
title(xlab = "Grain size (microns)")
```