Herbaceous Biomass OSD sorting

Courtney Meier

10/14/2016

Background

- Sorting OSD from current-year clipped biomass requires an extraordinary amount of time in D09 on a per clip strip basis.
- Kentucky Bluegrass 'mat' is highly intertwined current-year and OSD; like matted hair according to Andrea A. This means that sorting essentially must be done blade by blade in the laboratory following collection in the field.
- Goal here is to determine a data-driven cap on OSD sort time. Because total sort time is quite variable from clip strip to clip strip, would like to know the following:
  - What % of clip/sort events are X% complete in Y minutes?
  - Preliminarily, aim for 95% of plots at 90% complete
- Data source:
  - D09 timed sorting of OSD in 15 or 30 minute intervals for, and weighed total mass of OSD sorted at each interval.
  - Enables analysis of % of cumulative OSD sorted through time on a per plot basis.

```
if (file.exists(
  '/Users/cmeier/Documents/neonScienceDocs/gitRepositories')){
  wdir <- "~/Documents/neonScienceDocs/gitRepositories/neonPlantSampling/hbpSampling/sampleSortTime"
}

setwd(wdir)

clipSort <- read.xlsx("hbpSampleSorting.xlsx", sheetIndex = 1, header = TRUE)
clipSort <- clipSort %>% select(-NA.) %>% filter(osdMass!="NA")
```

Data frame manipulation

- Add clipID column
- Add cumulative sum of sorted osdMass and cumulative percent osdMass

```
clipID <- paste(clipSort$plotID, clipSort$clipCellNumber, sep = "_")
clipSort$clipID <- clipID

# Calculate cumulative sum and percent of osdMass per clipID, and add to original 'clipSort' dataframe
uniqueClip <- unique(clipSort$clipID)
cumulativeOSD <- NULL
cumulativePercent <- NULL
for (i in 1:length(unique(clipSort$clipID))){
  temp <- filter(clipSort, clipID==uniqueClip[i])
  cumSumTemp <- cumsum(temp$osdMass)
  cumPercentTemp <- round((cumSumTemp/max(cumSumTemp))*100)
  cumulativeOSD <- append(cumulativeOSD, cumSumTemp)
  cumulativePercent <- append(cumulativePercent, cumPercentTemp)
}

clipSort$cumulativeOSD <- cumulativeOSD
clipSort$cumulativePercent <- cumulativePercent
```

Calculations

- At 30 min time intervals, calculate the % of plots with 90% of sorting complete

```
theTimes <- seq(from=30, to=330, by=30)

# Calculations for 80% complete OSD sorting
percentPlots <- NULL
plotNum80Complete <- NULL
plotPercent80Complete <- NULL

for (i in 1:length(theTimes)){
  thePlots <- clipSort %>% filter(time <= theTimes[i] & cumulativePercent >= 80)
  numPlots <- length(unique(thePlots$clipID))
  percentPlots <- round((length(unique(thePlots$clipID))/length(uniqueClip))*100, digits = 1)
  plotNum80Complete <- append(plotNum80Complete, numPlots)
  plotPercent80Complete <- append(plotPercent80Complete, percentPlots)
}

# Calculations for 90% complete OSD sorting
percentPlots <- NULL
plotNum90Complete <- NULL
plotPercent90Complete <- NULL

for (i in 1:length(theTimes)){
  thePlots <- clipSort %>% filter(time <= theTimes[i] & cumulativePercent >= 90)
  numPlots <- length(unique(thePlots$clipID))
  percentPlots <- round((length(unique(thePlots$clipID))/length(uniqueClip))*100, digits = 1)
  plotNum90Complete <- append(plotNum90Complete, numPlots)
  plotPercent90Complete <- append(plotPercent90Complete, percentPlots)
}

clipResults <- data_frame(theTimes, plotNum80Complete, plotPercent80Complete, plotNum90Complete, plotPe
```

Summary Table

```
clipResults %>% kable
```

theTimes	plotNum80Complete	plotPercent80Complete	plotNum90Complete	plotPercent90Complete
30	0	0.0	0	0.0
60	9	28.1	4	12.5
90	17	53.1	11	34.4
120	22	68.8	20	62.5
150	27	84.4	23	71.9

theTimes	plotNum80Complete	plotPercent80Complete	plotNum90Complete	plotPercent90Complete
180	28	87.5	27	84.4
210	30	93.8	27	84.4
240	30	93.8	29	90.6
270	30	93.8	30	93.8
300	32	100.0	30	93.8
330	32	100.0	32	100.0