The DART Matrix

Daniel Spakowicz

4/24/2017

This is the start of Dan's effort on the final project for CBB750. The intention of this document is to perform some exploratory data analysis and to create summary slides for the presentation.

```
# Read in data
x <- read.csv("TheDartMatrix-deidentified.csv", as.is = TRUE)

# Read in variable key that defines classes
key <- read.csv("variable_key.csv")
levels(key$class)
```

```
## [1] "character" "factor"    "integer"   "numeric"
```

```
# Set classes
factors <- grep("factor", key$class)
char <- grep("character", key$class)
int <- grep("integer", key$class)
num <- grep("numeric", key$class)

x[,factors] <- lapply(x[factors], factor)
x[,char] <- lapply(x[char], as.character)
x[,int] <- lapply(x[int], as.integer)
x[,num] <- lapply(x[num], as.numeric)

# Tidy up
rm(factors)
rm(char)
rm(int)
rm(num)
```

Let's look at a quick table of when `fits_matrix` is called and when the DART consult is called.

```
table(x$fits_matrix)
```

0 1 322 353

```
print(xtable(table(fits_matrix = x$fits_matrix, consult_dart = x$consult_dart)),
      comment = FALSE)
```

	0	1	2	3
0	312	9	1	0
1	221	122	0	10

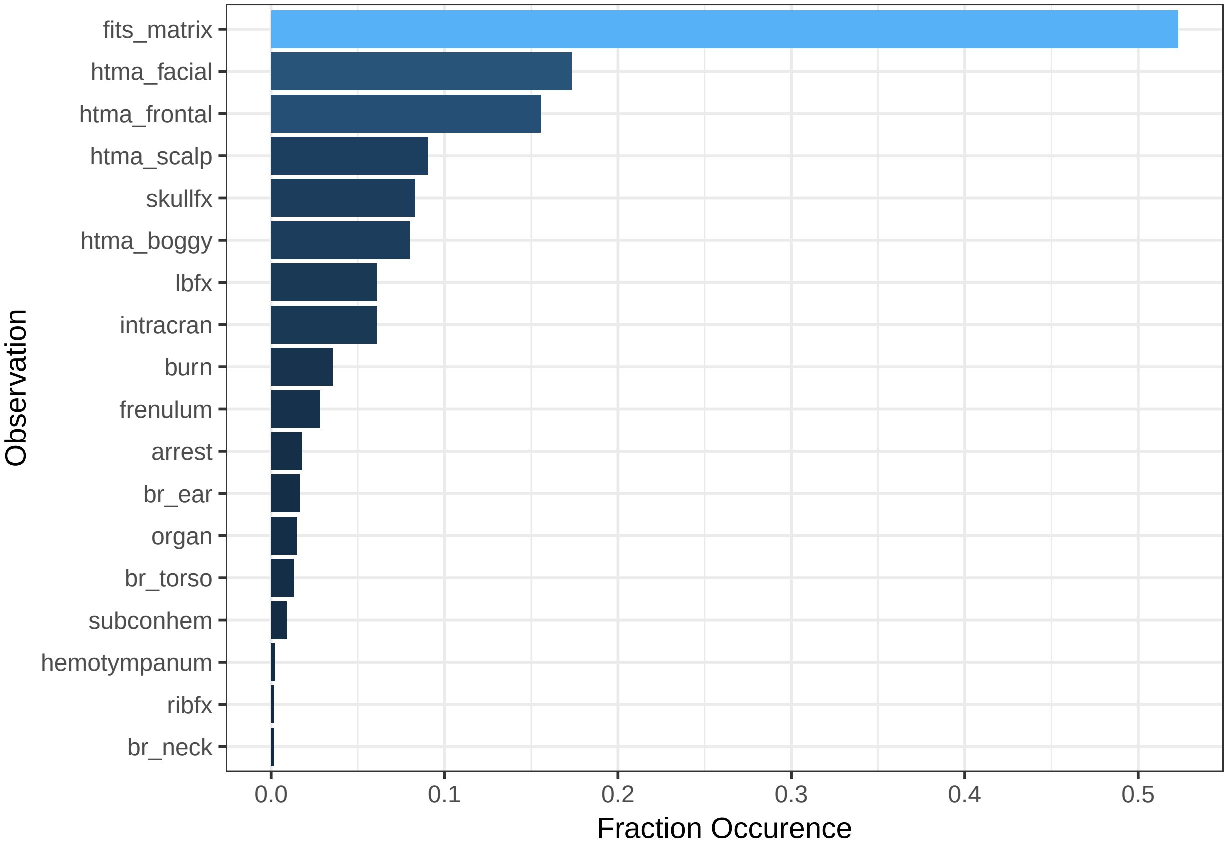
```
table(fits_matrix = x$fits_matrix, consult_dart = x$consult_dart)
```

```
       consult_dart
```

fits_matrix 0 1 2 3 0 312 9 1 0 1 221 122 0 10

Now I'll look at the frequency of the predictor variables.

```
mat <- data.frame(lapply(x[,5:22], as.numeric))
means <- data.frame(lapply(mat, function(x) mean(x-1, na.rm = TRUE)))
tmeans <- data.frame(var = names(means), perc_occurrence = t(means))
ggplot(tmeans, aes(x = reorder(var, perc_occurrence), y = perc_occurrence)) +
  geom_bar(stat = "identity", aes(fill = perc_occurrence)) +
  coord_flip() +
  labs(y = "Fraction Occurence",
       x = "Observation") +
  theme_bw() +
  theme(legend.position = "none") +
  ggsave("var_fracOccurrence.png", height = 4, width = 7.5)
```



```
# Tidy up
rm(mat)
rm(means)
# rm(tmeans)
```

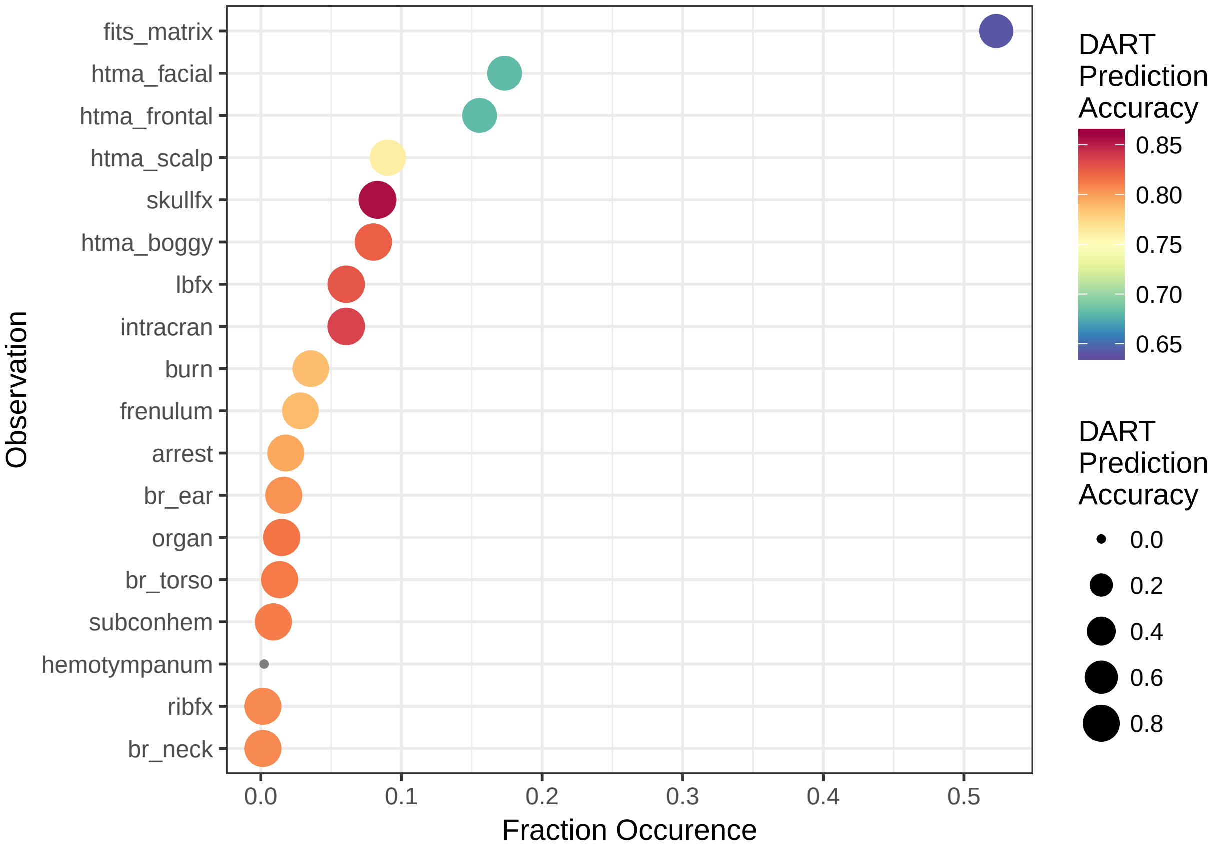
I'll try coloring these bars by the predictive accuracy

```
# Collapse 2 into 1 and 3 into 0
dart_collapsed <- x$consult_dart
for (i in 1:nrow(x)) {
  if (dart_collapsed[i] == 2){
    dart_collapsed[i] <- 1
  }
  if (dart_collapsed[i] == 3){
    dart_collapsed[i] <- 0
  }
}
dart_collapsed <- factor(dart_collapsed)

# Find the predictive accuracy of each var
predacc <- apply(x[,5:22], 2, function(x) mean(x == dart_collapsed))
predacc[is.na(predacc)] <- 0

myPalette <- colorRampPalette(rev(brewer.pal(11, "Spectral")))
sf <- scale_color_gradientn(colors = myPalette(100), limits=c(0.64,0.86))

ggplot(tmeans, aes(x = reorder(var, perc_occurrence), y = perc_occurrence)) +
  geom_point(stat = "identity", aes(color = predacc, size = predacc)) +
  coord_flip() +
  labs(y = "Fraction Occurence",
       x = "Observation",
       color = "DART\nPrediction\nAccuracy",
       size = "DART\nPrediction\nAccuracy") +
  theme_bw() +
  sf +
  ggsave("var_fracOccurrence_predAcc.png", height = 4, width = 7.5)
```



```
# Garbage collection
rm(myPalette)
rm(predacc)
rm(sf)
rm(tmeans)
rm(dart_collapsed)
```

```
# Convert notes into a list of word strings
notes <- vector(mode = "list", length = nrow(x))
for (i in 1:nrow(x)) {
  notes[[i]] <- unlist(strsplit(as.character(x$NOTE_TEXT[i]), split ="\\N", fixed = TRUE))
}

# Search for DART
dart <- as.numeric(unlist(lapply(notes, function(x) any(grep("[Dd][Aa][Rr][Tt]", x)))))

# Confusion matrix for using the presence of the word DART as a predictor of dart_consult
table(dart, consult_dart = x$consult_dart)
```

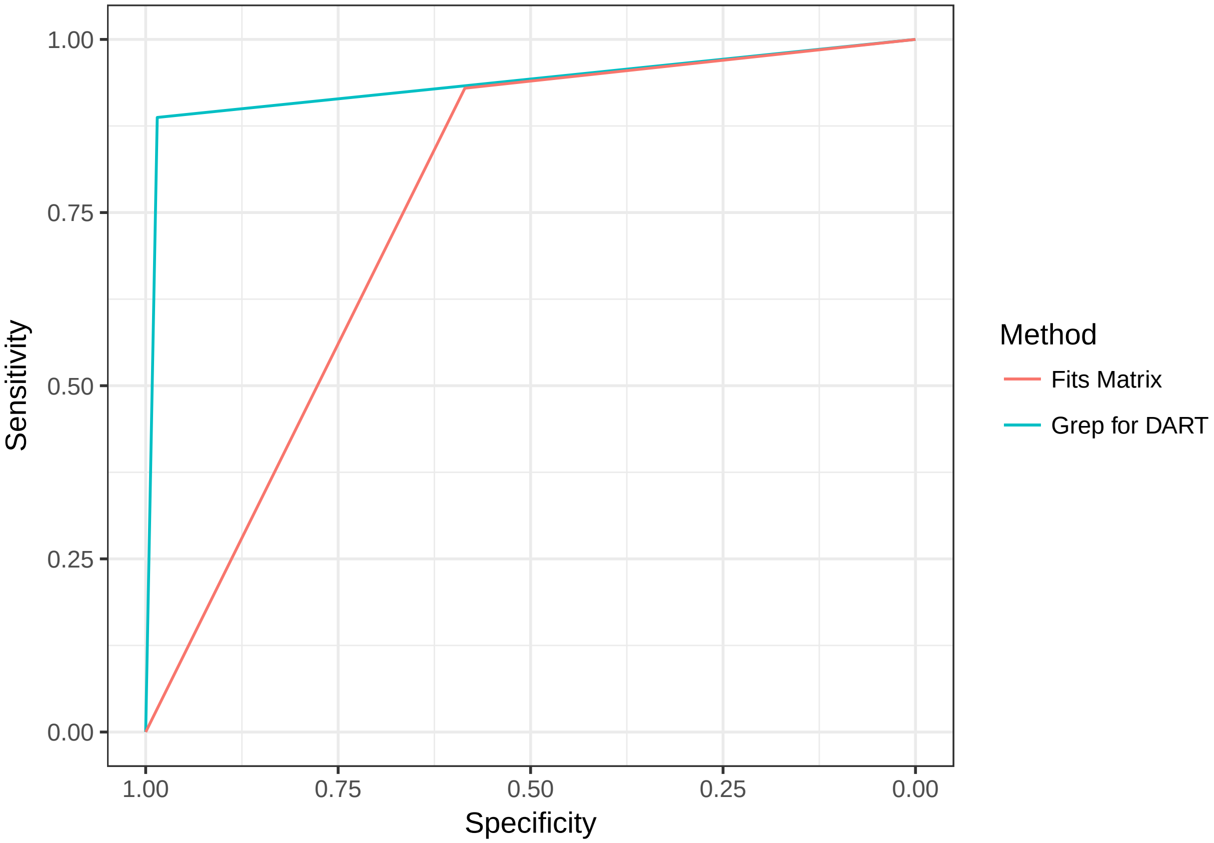
```
##      consult_dart
## dart   0   1   2   3
##    0 525   9   0   7
##    1   8 122   1   3
```

```
# Convert to bindary of called or not called (collapse 2 and 3 into 1)
consult_dart_binary <- ifelse(as.numeric(as.character(x$consult_dart)) >= 1, 1, 0)

# Create ROC curve for predicting dart call by fits_matrix vs DART grep
roc_fits <- roc(response = consult_dart_binary, predictor = as.numeric(as.character(x$fits_matrix)))
roc_dartgrep <- roc(response = consult_dart_binary, predictor = dart)

plotdf <- data.frame(dart_sp = roc_dartgrep$specificities, dart_sen = roc_dartgrep$sensitivities,
                     mat_sp = roc_fits$specificities, mat_sen = roc_fits$sensitivities)

ggplot(plotdf, aes(plotdf)) +
  geom_line(aes(x = dart_sp, y = dart_sen, color = "red")) +
  geom_line(aes(x = mat_sp, y = mat_sen, color = "blue")) +
  scale_x_reverse() +
  labs(x = "Specificity",
       y = "Sensitivity") +
  theme_bw() +
  scale_color_discrete(name="Method",
                         breaks=c("blue", "red"),
                         labels=c("Fits Matrix", "Grep for DART")) +
  ggsave("grepDART_roc.png", height = 4, width = 7.5)
```



```
# Error using grep
1- mean(consult_dart_binary == dart)
```

```
## [1] 0.03555556
```

```
# Error just using if it fits the matrix
1- mean(x$fits_matrix == consult_dart_binary)
```

```
## [1] 0.3422222
```