Classifier Performance

Rachael Caelie Aikens

May 1, 2018

Functions for loading and preprocessing data

```
labs_done <- c("TNI", "PTT", "PHOS", "MGN", "LACWB", "HEPAR", "CRP", "K", "FER", "PLTS", "HCTX", "LIPS"
percents <- 1:5*10
n_models <- 6
```

```
# construct from filepath and load
load_data <- function(lab, percents = 1:5, n_models = 6){
  paths <- paste("LAB", lab, "/change_percent_0", percents/10, "/LAB", lab, "-change-prediction-report.
                 sep = "")
  report <- rbindlist(lapply(paths, read.csv,
                             sep = "\t",
                             comment.char = "#",
                             stringsAsFactors = FALSE))
  return(report)
}
```

```
# parse dictionary string and return number of cases
get_n_cases <- function(y_dict){
  str_list <- gsub('.{1}$', '', strsplit(y_dict, " ")[[1]])
  case <- as.numeric(str_list[4])
  return(case)
}

# parse dictionary string and return number of control
get_n_controls <- function(y_dict){
  str_list <- gsub('.{1}$', '', strsplit(y_dict, " ")[[1]])
  control <- as.numeric(str_list[2])
  return(control)
}

# preprocess raw loaded data
preprocess_data <- function(report, lab, percents = 1:5, n_models = 6){
  report$percent_change <-  rep(percents, each = n_models)
  report$LAB <- lab
  report$test_cases <- sapply(report$y_test.value_counts..,
                                      function(x) get_n_cases(x))
  report$test_controls <- sapply(report$y_test.value_counts..,
                                      function(x) get_n_controls(x))
  return(report)
}

# run load and preprocessing and return processed result data
load_and_process <- function(lab, percents = 1:5, n_models = 6){
  raw <- load_data(lab, percents, n_models)
  return(preprocess_data(raw, lab, percents, n_models))
```

```
}
```

Analysis

One Lab

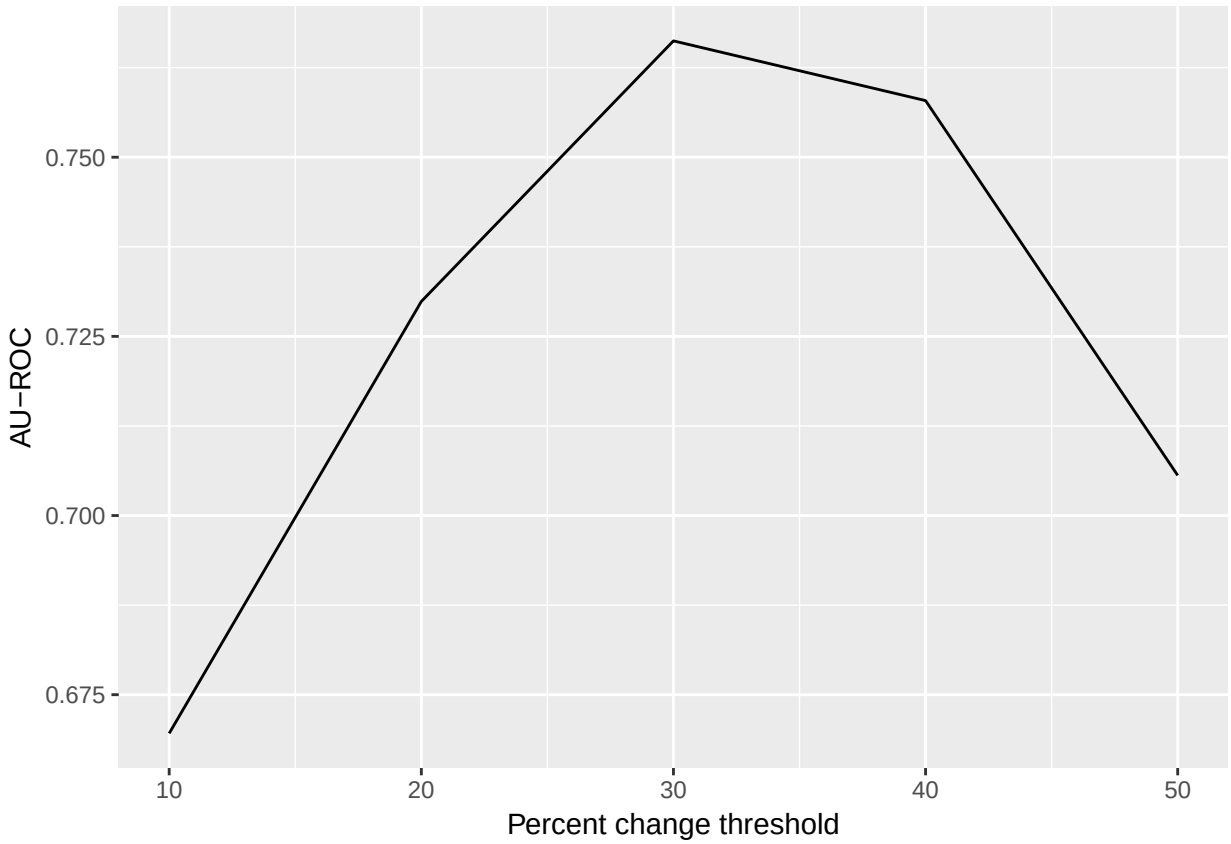
Here is an example of analyses we could do for a single lab test. Here, we use TNI

```
full_report <- load_and_process("FER", percents)
```

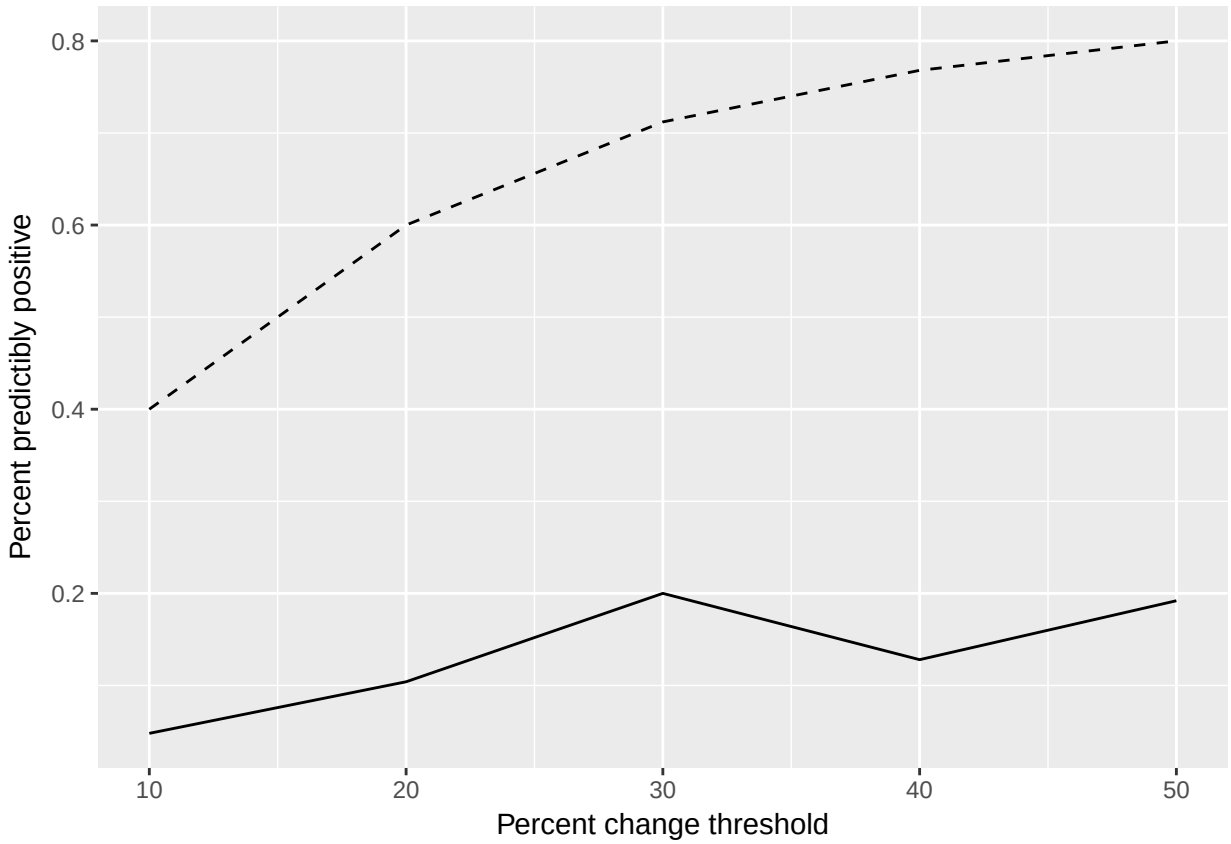
```
report_summary <- full_report %>%
  group_by(percent_change) %>%
  summarize(max_roc = max(roc_auc),
            max_percent_predictability = max(percent_predictably_positive),
            test_cases = first(test_cases),
            test_controls = first(test_controls))
report_summary
```

```
## # A tibble: 5 x 5
##   percent_change max_roc max_percent_predictabil~ test_cases test_controls
##            <dbl>   <dbl>                   <dbl>      <dbl>         <dbl>
## 1             10   0.670                   0.048         50            75
## 2             20   0.730                   0.104         75            50
## 3             30   0.766                   0.2           89            36
## 4             40   0.758                   0.128         96            29
## 5             50   0.706                   0.192        100            25
```

```
ggplot(report_summary, aes(x = percent_change, y = max_roc)) +
  geom_line()+
  labs(x = "Percent change threshold", y = "AU-ROC")
```




```
ggplot(report_summary, aes(x = percent_change, y = max_percent_predictability)) +
  geom_line() +
  geom_line(aes(x = percent_change, y = test_cases/(test_cases+test_controls)), linetype = "dashed") +
  #geom_line(aes(x = percent_change, y = max_percent_predictability*test_cases/(test_cases+test_control
  labs(x = "Percent change threshold", y = "Percent predictibly positive")
```

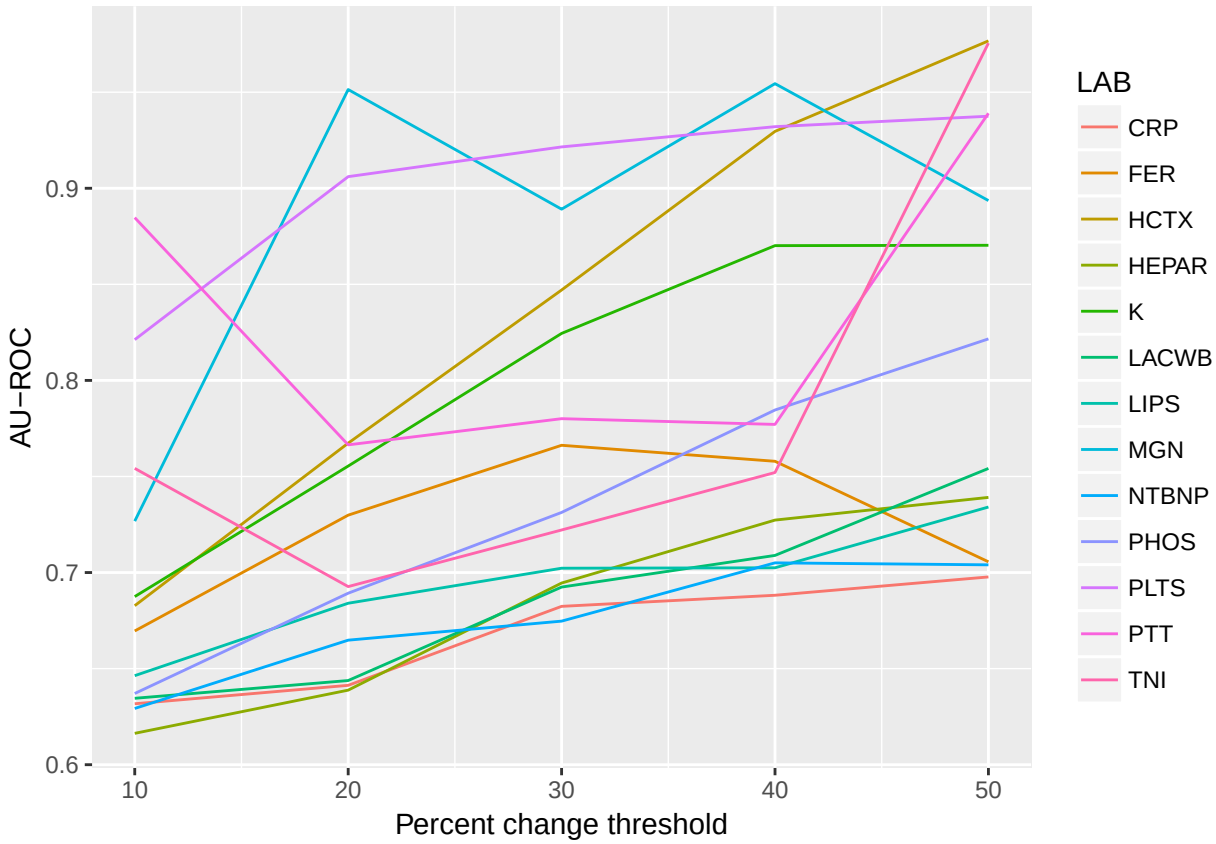


Many Labs

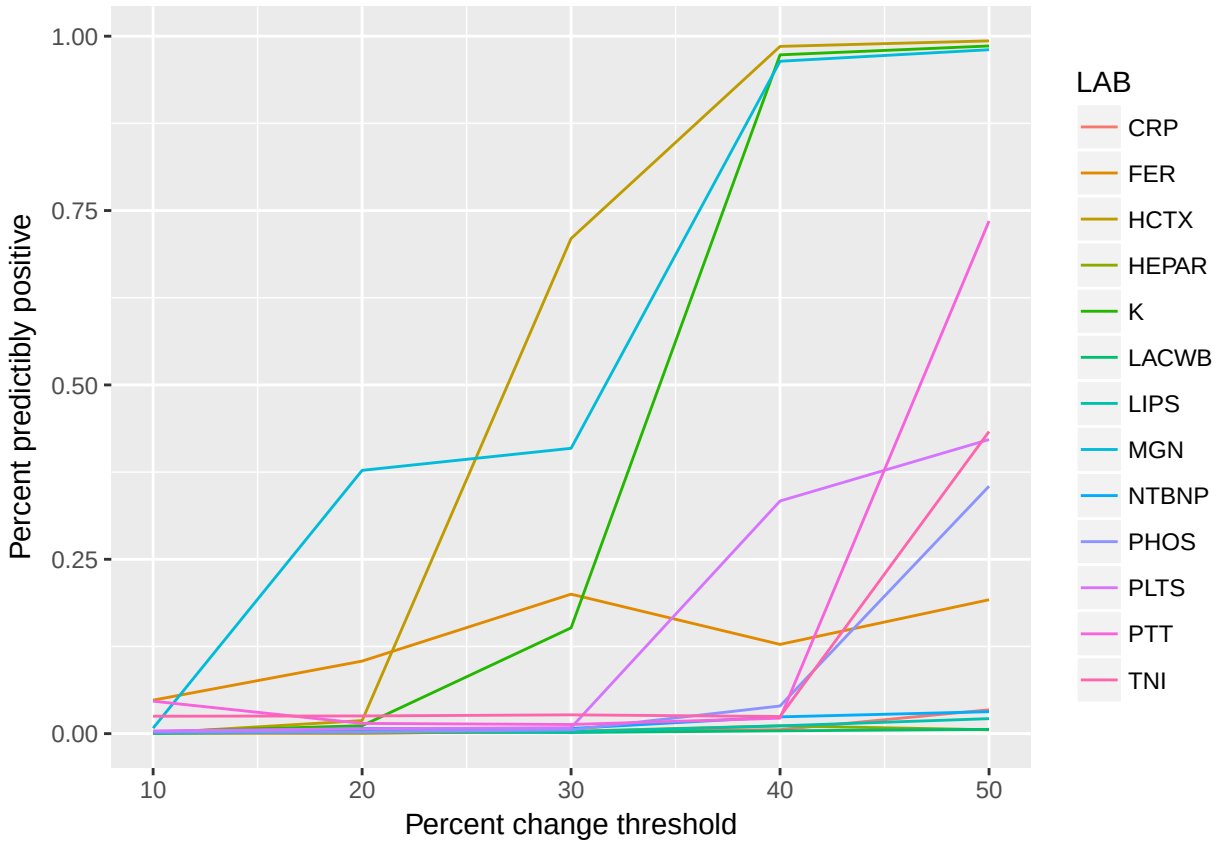
```
full_data <- rbindlist(lapply(labs_done, load_and_process, percents = percents))
```

```
report_summary <- full_data %>%
  group_by(percent_change, LAB) %>%
  summarize(max_roc = max(roc_auc),
            max_percent_predictability = max(percent_predictably_positive),
            test_cases = first(test_cases),
            test_controls = first(test_controls))
```

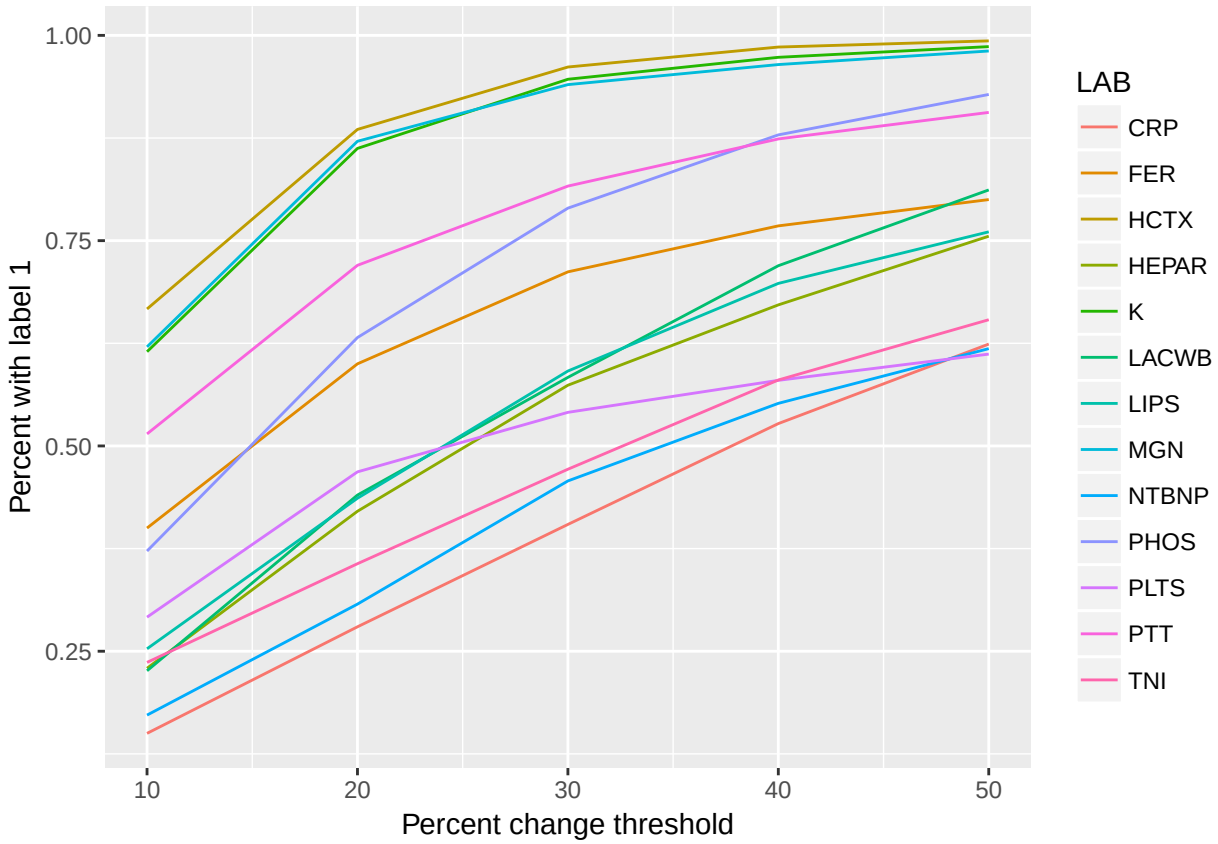
```
ggplot(report_summary, aes(x = percent_change, y = max_roc, group = LAB, color = LAB)) +
  geom_line()+
  labs(x = "Percent change threshold", y = "AU-ROC")
```




```
ggplot(report_summary, aes(x = percent_change, y = max_percent_predictability, group = LAB, color = LAB)
  geom_line() +
  #geom_line(aes(x = percent_change, y = test_cases/(test_cases+test_controls)), linetype = "dashed") +
  #geom_line(aes(x = percent_change, y = max_percent_predictability*test_cases/(test_cases+test_control
  labs(x = "Percent change threshold", y = "Percent predictibly positive")
```




```
ggplot(report_summary, aes(x = percent_change, y = test_cases/(test_cases+test_controls), group = LAB, c
  geom_line() +
  #geom_line(aes(x = percent_change, y = test_cases/(test_cases+test_controls)), linetype = "dashed") +
  #geom_line(aes(x = percent_change, y = max_percent_predictability*test_cases/(test_cases+test_control
  labs(x = "Percent change threshold", y = "Percent with label 1")
```


1.00
0.75
0.50
0.25
Percent with label 1
10
20
30
40
50
Percent change threshold
LAB
CRP
FER
HCTX
HEPAR
K
LACWB
LIPS
MGN
NTBNP
PHOS
PLTS
PTT
TNI