SIAMCAT_iletis_080823

Suet Li Hooi

2023-12-07

```
### ASIAN and WESTERN model
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.2     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.2     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(ggplot2)
library(phyloseq)
library(reshape2)
```

```
##
## Attaching package: 'reshape2'
##
## The following object is masked from 'package:tidyr':
##
##     smiths
```

```
library(vegan)
```

```
## Loading required package: permute
## Loading required package: lattice
## This is vegan 2.6-4
```

```
library(ggsci)
library(ggsignif)
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'
##
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
library(microbiome)
```

```
##
## microbiome R package (microbiome.github.com)
##
##
##
##  Copyright (C) 2011-2022 Leo Lahti,
##     Sudarshan Shetty et al. <microbiome.github.io>
##
##
## Attaching package: 'microbiome'
##
## The following object is masked from 'package:vegan':
##
##     diversity
##
## The following object is masked from 'package:ggplot2':
##
##     alpha
##
## The following object is masked from 'package:base':
##
##     transform
```

```
library(microbiomeMarker)
```

```
## Registered S3 method overwritten by 'gplots':
##   method         from
##   reorder.factor DescTools
##
## Attaching package: 'microbiomeMarker'
##
## The following objects are masked from 'package:microbiome':
##
##     abundances, aggregate_taxa
##
## The following object is masked from 'package:phyloseq':
##
##     plot_heatmap
```

```
library(SIAMCAT)
```

```
## Loading required package: mlr3
```

```
## Warning: package 'mlr3' was built under R version 4.3.1
```

```
##
## Attaching package: 'SIAMCAT'
##
## The following object is masked from 'package:microbiome':
##
##     meta
```

```
setwd("~/AMILI_2022/hpc2_projects/vsl3_analysis/ILETIS_NEW_020823")
```

```
#upload asian metadata
metadata_sg <- read.csv('./asian_iletis/leftover_healthy_amili89.csv') %>%
  select(Health_status, Age, Gender, BMI, X16S_AnalysisID) %>%
  dplyr::rename(Sample = X16S_AnalysisID) %>%
  mutate(BMI = as.numeric(BMI), Country = "Singapore")
```

```
## Warning: There was 1 warning in `mutate()`.
## i In argument: `BMI = as.numeric(BMI)`.
## Caused by warning:
## ! NAs introduced by coercion
```

```
metadata_chinese <- read.delim('./asian_iletis/PRJNA835157_chinese.txt',
                               sep = ',') %>%
  filter(Isolation_Source %in% "feces") %>%
  select(Run, Host_age, host_sex) %>%
  mutate(Health_status = "IBD", BMI = NA, Country = "China") %>%
  dplyr::rename(Sample = Run, Age = Host_age, Gender = host_sex)

metadata_korean <- read.delim('./asian_iletis/PRJNA968150_korean.txt',
                              sep = ',') %>%
  filter(Sample.Name %in% "IBD microbiome") %>%
  select(Run) %>%
  dplyr::rename(Sample = Run) %>%
  mutate(Health_status = "IBD", Age = NA, Gender = NA, BMI = NA,
         Country = "Korea")
```

```
#merge all asian metadatas
metadata_asian <- bind_rows(metadata_chinese, metadata_korean, metadata_sg)
```

```
#upload asian edge table from HPC
Edge_asian <- read.csv('./asian_iletis/iletis_edge_bind_asian.csv') %>%
  select(-X) %>%
  mutate(V1 = gsub(".16S.exp.", "", V1)) %>%
  filter(V1 %in% metadata_asian$Sample) %>%
   distinct(V1, .keep_all = TRUE) %>%
  column_to_rownames('V1') %>%
  t() %>%
  as.data.frame() %>%
   replace(is.na(.), 0)
```

```
#upload asian taxa table

Taxa_asian <- read.csv('./asian_iletis/iletis_taxa_bind_asian.csv') %>%
  setNames(.[1,]) %>%
  .[-1,]

colnames(Taxa_asian)[1] <- "Index"
colnames(Taxa_asian)[2] <- "Edge"

rownames(Taxa_asian) <- NULL
```

```
Taxa_asian <- Taxa_asian %>%
  select(-Index) %>%
  filter(!duplicated(Edge)) %>%
  filter(!is.na(Edge)) %>%
  column_to_rownames("Edge") %>%
  dplyr::rename(Kingdom = superkingdom, Phylum = phylum,
                Class = class, Order = order, Family = family,
                Genus = genus, Species = species) %>%
  select(-clade, -strain, -taxon) %>%
  mutate_all(na_if, "") %>%
  replace_na(list(Order = 'Unclassified', Family = 'Unclassified',
                  Genus = 'Unclassified', Species = 'Unclassified'))
```

```
#phyloseq
OTU <- otu_table(as.matrix(Edge_asian), taxa_are_rows = T)

TAXA <- tax_table(as.matrix(Taxa_asian))

metadata_asian <- column_to_rownames(metadata_asian, "Sample")

SAMPLE <- sample_data(metadata_asian)

ps.asian <- phyloseq(OTU, TAXA, SAMPLE)

#transform to relative abundance
psr.asian  = transform_sample_counts(ps.asian, function(x) x / sum(x) )
psr.asian
```

```
## phyloseq-class experiment-level object
## otu_table()   OTU Table:         [ 3749 taxa and 213 samples ]
## sample_data() Sample Data:       [ 213 samples by 5 sample variables ]
## tax_table()   Taxonomy Table:    [ 3749 taxa by 7 taxonomic ranks ]
```

```
#remove species with abundances below 0.0001
psr.asian.filt <- phyloseq::genefilter_sample(psr.asian,
                                              filterfun_sample
                                              (function(x) x >= 0.0001))

psr.asian.filt <- prune_taxa(psr.asian.filt, psr.asian)
psr.asian.filt
```

```
## phyloseq-class experiment-level object
## otu_table()   OTU Table:         [ 2410 taxa and 213 samples ]
## sample_data() Sample Data:       [ 213 samples by 5 sample variables ]
## tax_table()   Taxonomy Table:    [ 2410 taxa by 7 taxonomic ranks ]
```

```
#readjust input dataframes to create siamcat object
Taxa_sc <- read.csv('iletis_taxa_bind.csv') %>%
  setNames(.[1,]) %>%
  .[-1,]
```

```
colnames(Taxa_sc)[1] <- "Index"
colnames(Taxa_sc)[2] <- "Edge"

rownames(Taxa_sc) <- NULL

Taxa_sc <- Taxa_sc %>%
  select(-Index) %>%
  filter(!duplicated(Edge)) %>%
  filter(!is.na(Edge)) %>%
  column_to_rownames("Edge") %>%
  dplyr::rename(Kingdom = superkingdom, Phylum = phylum, Class = class, Order = order, Family = family,
  mutate_all(na_if, "") %>%
  replace_na(list(Order = 'Unclassified', Family = 'Unclassified', Genus = 'Unclassified', Species = 'U
  rownames_to_column('Edge') %>%
  mutate(Edge2 = Edge, Taxon2 = taxon) %>%
  unite(Taxa, Edge2, Taxon2, sep = ' | ') %>%
  column_to_rownames('Taxa')
```

```
#upload metadata
metadata_updated165 <- read.csv('metadata_ibd_healthy_updated165.csv') %>%
  mutate(X16S_AnalysisID = gsub("-", "", X16S_AnalysisID))
```

```
#upload edge table from HPC
Edge <- read.csv('iletis_edge_bind.csv') %>%
  select(-X, -Sample) %>%
  mutate(V1 = gsub(".16S.exp.", "", V1)) %>%
  filter(V1 %in% metadata_updated165$X16S_AnalysisID) %>%
  column_to_rownames('V1') %>%
  t() %>%
  as.data.frame() %>%
   replace(is.na(.), 0)
```

```
#upload taxa table

Taxa <- read.csv('iletis_taxa_bind.csv') %>%
  setNames(.[1,]) %>%
  .[-1,]

colnames(Taxa)[1] <- "Index"
colnames(Taxa)[2] <- "Edge"

rownames(Taxa) <- NULL

Taxa <- Taxa %>%
  select(-Index) %>%
  filter(!duplicated(Edge)) %>%
  filter(!is.na(Edge)) %>%
  column_to_rownames("Edge") %>%
  dplyr::rename(Kingdom = superkingdom, Phylum = phylum, Class = class,
                Order = order, Family = family,
                Genus = genus, Species = species) %>%
  select(-clade, -strain, -taxon) %>%
  mutate_all(na_if, "") %>%
```

```
  replace_na(list(Order = 'Unclassified',
                  Family = 'Unclassified',
                  Genus = 'Unclassified',
                  Species = 'Unclassified'))
```

```
#phyloseq
OTU <- otu_table(as.matrix(Edge), taxa_are_rows = T)

TAXA <- tax_table(as.matrix(Taxa))

metadata_updated165 <- column_to_rownames(metadata_updated165,
                                          "X16S_AnalysisID")

SAMPLE <- sample_data(metadata_updated165)

ps <- phyloseq(OTU, TAXA, SAMPLE)
```

```
#siamcat for species level --training set
Edge_filt <- as(otu_table(ps), 'matrix') %>%
  as.data.frame() %>%
  rownames_to_column('Edge') %>%
  merge(Taxa_sc, by = "Edge") %>%
  select(-Edge, -Kingdom, -Phylum, -Class, -Order, -Family,
         -Genus, -clade, -strain, -taxon)
```

```
psrmelt.species <- aggregate(. ~ Species, Edge_filt, sum)

psrmelt.species <- column_to_rownames(psrmelt.species, 'Species')
```

```
#siamcat for species level
Edge_filt.asian <- as(otu_table(ps.asian), 'matrix') %>%
  as.data.frame() %>%
  rownames_to_column('Edge') %>%
  merge(Taxa_sc, by = "Edge") %>%
  select(-Edge, -Kingdom, -Phylum, -Class, -Order, -Family,
         -Genus, -clade, -strain, -taxon)
```

```
psrmelt.species.asian <- aggregate(. ~ Species, Edge_filt.asian, sum)

psrmelt.species.asian <- column_to_rownames(psrmelt.species.asian, 'Species')
```

```
psrmelt.species <- psrmelt.species %>%
  filter(rownames(.) %in% rownames(psrmelt.species.asian))
```

```
psrmelt.species <- prop.table(as.matrix(psrmelt.species), 2)

psrmelt.species.asian <- prop.table(as.matrix(psrmelt.species.asian), 2)
```

```
#siamcat for Species level (Abundances summed from total of each species group)
set.seed(123)
sc.obj.species.asian <- siamcat(feat= psrmelt.species.asian,
```

```
    label='Health_status', case='IBD',
    meta=metadata_asian)
```

```
## + starting create.label

## Label used as case:
##    IBD
## Label used as control:
##    Healthy

## + finished create.label.from.metadata in 0.02 s

## + starting validate.data

## +++ checking overlap between labels and features

## + Keeping labels of 213 sample(s).

## + Removed 17 samples from the label object...

## +++ checking sample number per class

## +++ checking overlap between samples and metadata

## + finished validate.data in 0.07 s
```

```
show(sc.obj.species.asian)
```

```
## siamcat-class object
## label()                Label object:         87 Healthy and 126 IBD samples
##
## contains phyloseq-class experiment-level object @phyloseq:
## phyloseq@otu_table()   OTU Table:            [ 873 taxa and 213 samples ]
## phyloseq@sam_data()    Sample Data:          [ 213 samples by 4 sample variables ]
```

```
psr.filt <- readRDS("~/AMILI_2022/hpc2_projects/vsl3_analysis/ILETIS_NEW_020823/psr.filt.rds")

metadata163 <- as(sample_data(psr.filt), 'data.frame')
```

```
metadata163.sc <- metadata163 %>%
  select(Health_status, Disease_category)
```

```
set.seed(123)
sc.obj.species <- siamcat(feat=psrmelt.species, meta=metadata163.sc,
    label='Health_status', case='IBD')
```

```
## + starting create.label
```

```
## Label used as case:
##    IBD
## Label used as control:
##    Healthy
```

```
## + finished create.label.from.metadata in    0 s
```

```
## + starting validate.data
```

```
## +++ checking overlap between labels and features
```

```
## + Keeping labels of 163 sample(s).
```

```
## +++ checking sample number per class
```

```
## +++ checking overlap between samples and metadata
```

```
## + finished validate.data in 0.01 s
```

```
set.seed(123)
sc.obj.species <- filter.features(
    sc.obj.species,
    filter.method = 'abundance',
    cutoff = 0.001,
    rm.unmapped = TRUE,
    verbose=2
)
```

```
## + starting filter.features
```

```
## +++ before filtering, the data have 873 features
```

```
## +++ removed 851 features corresponding to UNMAPPED reads
```

```
## +++ removed 501 features whose values did not exceed 0.001 in any sample (retaining 372)
```

```
## + finished filter.features in 0.01 s
```

```
set.seed(123)
sc.obj.species <- normalize.features(
    sc.obj.species,
    norm.method = "log.std",
    norm.param = list(log.n0 = 1e-06, sd.min.q = 0.1),
    verbose = 2
)
```

```
## + starting normalize.features
```

```
## +++ performing de novo normalization using the  log.std  method
```

```
## + feature sparsity before normalization: 66.93%

## +++ feature sparsity after normalization:      0 %

## + finished normalize.features in 0.01 s
```

```
set.seed(123)
sc.obj.species <-  create.data.split(
    sc.obj.species,
    num.folds = 5,
    num.resample = 5
)
```

```
## Features splitted for cross-validation successfully.
```

```
set.seed(123)
sc.obj.species <- train.model(
    sc.obj.species,
    method = "lasso"
)
```

```
## Trained lasso models successfully.
```

```
set.seed(123)
sc.obj.species <- make.predictions(sc.obj.species)
```

```
## Made predictions successfully.
```

```
sc.obj.species <-  evaluate.predictions(sc.obj.species)
```

```
## Evaluated predictions successfully.
```

```
set.seed(123)
sc.obj.species.asian <- normalize.features(sc.obj.species.asian,
    norm.param=norm_params(sc.obj.species),
    feature.type='original',
    verbose = 2)
```

```
## + starting normalize.features

## + normalizing original features

## + performing frozen  log.std  normalization using the supplied parameters

## + feature sparsity before normalization: 77.47%

## + feature sparsity after normalization:     0%

## + finished normalize.features in    0 s
```

```
set.seed(123)
sc.obj.species.asian <- make.predictions(
    siamcat = sc.obj.species,
    siamcat.holdout = sc.obj.species.asian,
    normalize.holdout = FALSE)
```

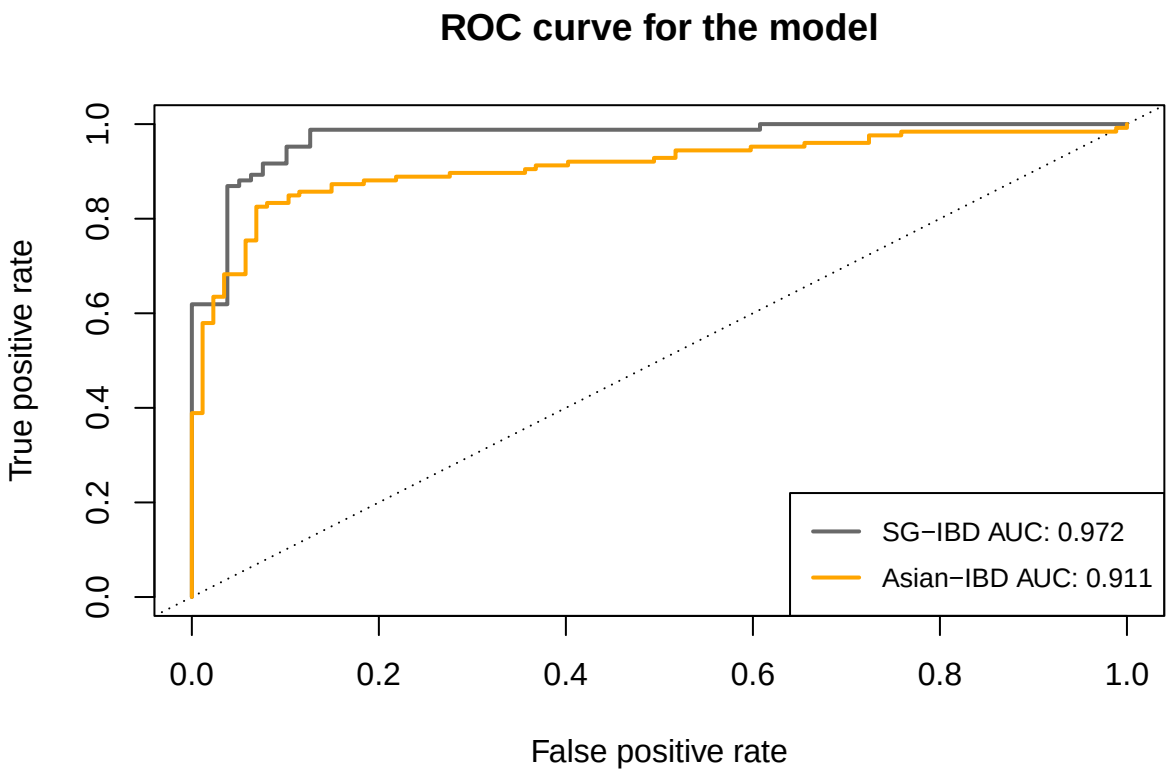
```
## Warning in make.external.predictions(siamcat.trained = siamcat,
## siamcat.external = siamcat.holdout, : WARNING: holdout set is not being
## normalized!
```

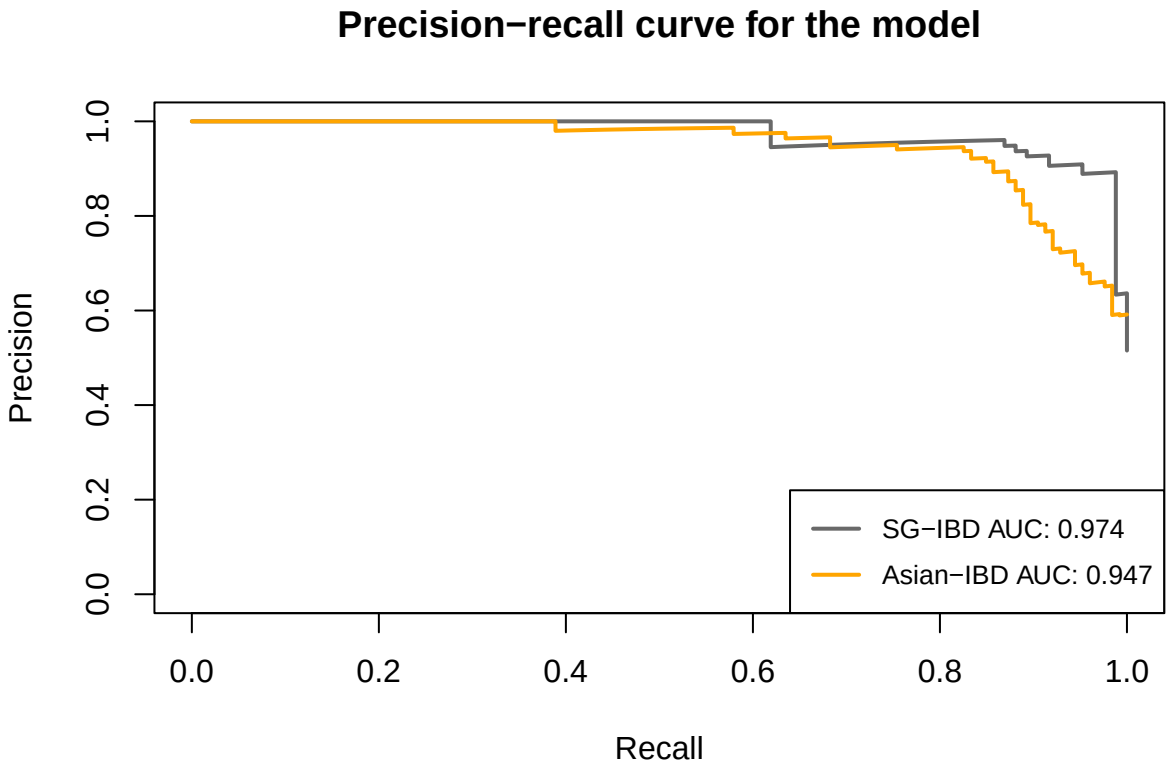
```
## Made predictions successfully.
```

```
set.seed(123)
sc.obj.species.asian <- evaluate.predictions(sc.obj.species.asian)
```

```
## Evaluated predictions successfully.
```

```
model.evaluation.plot('SG-IBD'=sc.obj.species,
    'Asian-IBD'=sc.obj.species.asian,
    colours=c('dimgrey', 'orange'))
```






```
model.interpretation.plot(sc.obj.species,
                          fn.plot = "interpretation.species.pdf",
    consens.thres = 0.5, limits = c(-3, 3), heatmap.type = 'zscore')
```

```
## Successfully plotted model interpretation plot to: interpretation.species.pdf
```

```
knitr::include_graphics("./interpretation.species.pdf")
```

```
#western validation cohort

metadata_west <- read.csv('./western_iletis/metadata_public_ibd_West_updated.csv') %>%
  select(-X, -Disease_category, -BioProject) %>%
  dplyr::rename(Sample = Run)
```

```
#upload asian edge table from HPC
Edge_western <- read.csv('./western_iletis/iletis_edge_bind_western.csv') %>%
  select(-X) %>%
  mutate(V1 = gsub(".16S.exp.", "", V1)) %>%
  filter(V1 %in% metadata_west$Sample) %>%
   distinct(V1, .keep_all = TRUE) %>%
  column_to_rownames('V1') %>%
  t() %>%
  as.data.frame() %>%
   replace(is.na(.), 0)
```

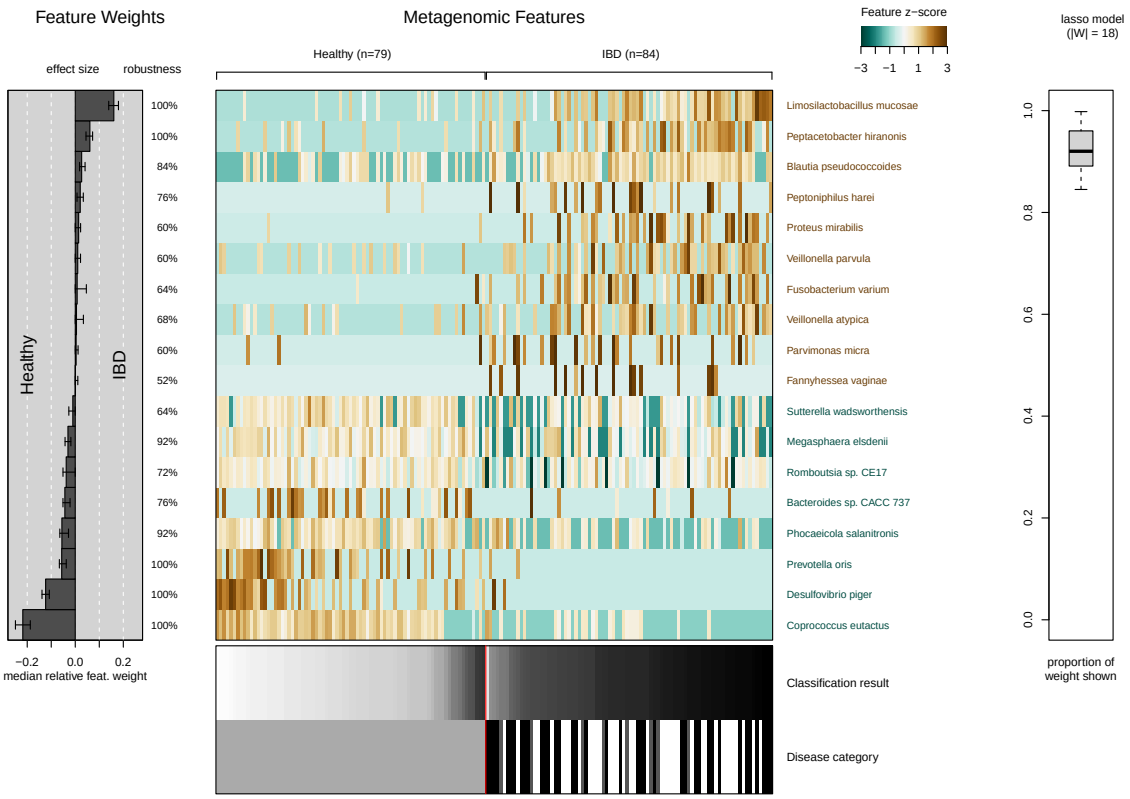


Figure 1: Important features for Singaporean IBD model

```
#upload western taxa table

Taxa_western <- read.csv('./western_iletis/iletis_taxa_bind_western.csv') %>%
  setNames(.[1,]) %>%
  .[-1,]

colnames(Taxa_western)[1] <- "Index"
colnames(Taxa_western)[2] <- "Edge"

rownames(Taxa_western) <- NULL

Taxa_western <- Taxa_western %>%
  select(-Index) %>%
  filter(!duplicated(Edge)) %>%
  filter(!is.na(Edge)) %>%
  column_to_rownames("Edge") %>%
  dplyr::rename(Kingdom = superkingdom, Phylum = phylum,
                Class = class, Order = order, Family = family,
                Genus = genus, Species = species) %>%
  select(-clade, -strain, -taxon) %>%
  mutate_all(na_if, "") %>%
  replace_na(list(Order = 'Unclassified',
                  Family = 'Unclassified',
                  Genus = 'Unclassified',
                  Species = 'Unclassified'))
```

```
#phyloseq
OTU <- otu_table(as.matrix(Edge_western), taxa_are_rows = T)

TAXA <- tax_table(as.matrix(Taxa_western))

metadata_west <- column_to_rownames(metadata_west, "Sample")

SAMPLE <- sample_data(metadata_west)

ps.west <- phyloseq(OTU, TAXA, SAMPLE)
```

```
#siamcat for species level
Edge_filt.west <- as(otu_table(ps.west), 'matrix') %>%
  as.data.frame() %>%
  rownames_to_column('Edge') %>%
  merge(Taxa_sc, by = "Edge") %>%
  select(-Edge, -Kingdom, -Phylum, -Class, -Order, -Family,
         -Genus, -clade, -strain, -taxon)
```

```
psrmelt.species.western <- aggregate(. ~ Species, Edge_filt.west, sum)

psrmelt.species.western <- column_to_rownames(psrmelt.species.western, 'Species')
```

```
#add 0 to species not available in western dataset from the asian and training set
psrmelt.species.western.new <- psrmelt.species.western %>%
  filter(rownames(.) %in% rownames(psrmelt.species))
```

```
#additional <- psrmelt.species %>%
 # filter(!rownames(.) %in% rownames(psrmelt.species.western.new)) %>%
 # mutate_all(~ 0) %>%
 # bind_rows(psrmelt.species.western.new) %>%
  #t() %>% as.data.frame() %>%
  #filter(rownames(.) %in% colnames(psrmelt.species.western.new)) %>%
  #replace(is.na(.), 0) %>%
  #t() %>% as.data.frame()

#write.csv(additional, 'additional.csv')
```

```
additional <- read.csv('additional.csv') %>%
  column_to_rownames('X')
psrmelt.species.western <- prop.table(as.matrix(additional), 2)
```

```
#siamcat for Species level (Abundances summed from total of each species group)
set.seed(123)
sc.obj.species.western <- siamcat(feat= psrmelt.species.western,
    label='Health_status', case='IBD',
    meta=metadata_west)
```

```
## + starting create.label

## Label used as case:
##    IBD
## Label used as control:
##    Healthy

## + finished create.label.from.metadata in    0 s

## + starting validate.data

## +++ checking overlap between labels and features

## + Keeping labels of 67 sample(s).

## + Removed 3 samples from the label object...

## +++ checking sample number per class

## +++ checking overlap between samples and metadata

## + finished validate.data in 0.03 s
```

```
show(sc.obj.species.western)
```

```
## siamcat-class object
## label()                Label object:         41 Healthy and 26 IBD samples
##
## contains phyloseq-class experiment-level object @phyloseq:
## phyloseq@otu_table()   OTU Table:            [ 873 taxa and 67 samples ]
## phyloseq@sam_data()    Sample Data:          [ 67 samples by 3 sample variables ]
```

```
set.seed(123)
sc.obj.species.western <- normalize.features(sc.obj.species.western,
    norm.param=norm_params(sc.obj.species),
    feature.type='original',
    verbose = 2)
```

```
## + starting normalize.features

## + normalizing original features

## + performing frozen  log.std  normalization using the supplied parameters

## + feature sparsity before normalization: 91.11%

## + feature sparsity after normalization:     0%

## + finished normalize.features in    0 s
```

```
set.seed(123)
sc.obj.species.western <- make.predictions(
    siamcat = sc.obj.species,
    siamcat.holdout = sc.obj.species.western,
    normalize.holdout = FALSE)
```

```
## Warning in make.external.predictions(siamcat.trained = siamcat,
## siamcat.external = siamcat.holdout, : WARNING: holdout set is not being
## normalized!

## Made predictions successfully.
```

```
set.seed(123)
sc.obj.species.western <- evaluate.predictions(sc.obj.species.western)
```

```
## Evaluated predictions successfully.
```

```
model.evaluation.plot('SG-IBD'=sc.obj.species,
    'Asian-IBD'=sc.obj.species.asian,
    'Western-IBD'=sc.obj.species.western,
    colours=c('dimgrey', 'orange', 'lightblue'),
    fn.plot = "AUC_curve_siamcat.pdf")
```

```
## Plotted evaluation of predictions successfully to: AUC_curve_siamcat.pdf
```

```
model.interpretation.plot(sc.obj.species, fn.plot = "interpretation.species.pdf",
    consens.thres = 0.5, limits = c(-3, 3), heatmap.type = 'zscore')
```

```
## Successfully plotted model interpretation plot to: interpretation.species.pdf
```

```r
#calculate from roc.all
sensitivity <- sc.obj.species@eval_data[["roc.all"]][[1]][["sensitivities"]] %>% as.data.frame()

mean(sensitivity$.)
```

```
## [1] 0.7275697
```

```r
specificity <- sc.obj.species@eval_data[["roc.all"]][[1]][["specificities"]] %>%
  as.data.frame() %>%
  rename_with(.cols = 1, ~"value") %>%
  mutate(value_1 = 1 - value)

mean(specificity$value)
```

```
## [1] 0.7419728
```

```r
#asian
sensitivity.asian <- sc.obj.species.asian@eval_data[["roc.all"]][[1]][["sensitivities"]] %>%
  as.data.frame() %>%
  rename_with(.cols = 1, ~"value")

mean(sensitivity.asian$value)
```

```
## [1] 0.6904762
```

```r
#asian
specificity.asian <- sc.obj.species.asian@eval_data[["roc.all"]][[1]][["specificities"]] %>% as.data.fra
  rename_with(.cols = 1, ~"value")

mean(specificity.asian$value)
```

```
## [1] 0.7301127
```

```r
#western
sensitivity.western <- sc.obj.species.western@eval_data[["roc.all"]][[1]][["sensitivities"]] %>%
  as.data.frame() %>%
  rename_with(.cols = 1, ~"value")

mean(sensitivity.western$value)
```

```
## [1] 0.620979
```

```r
#western
specificity.western <- sc.obj.species.western@eval_data[["roc.all"]][[1]][["specificities"]] %>% as.data
  rename_with(.cols = 1, ~"value")

mean(specificity.western$value)
```

```
## [1] 0.568071
```

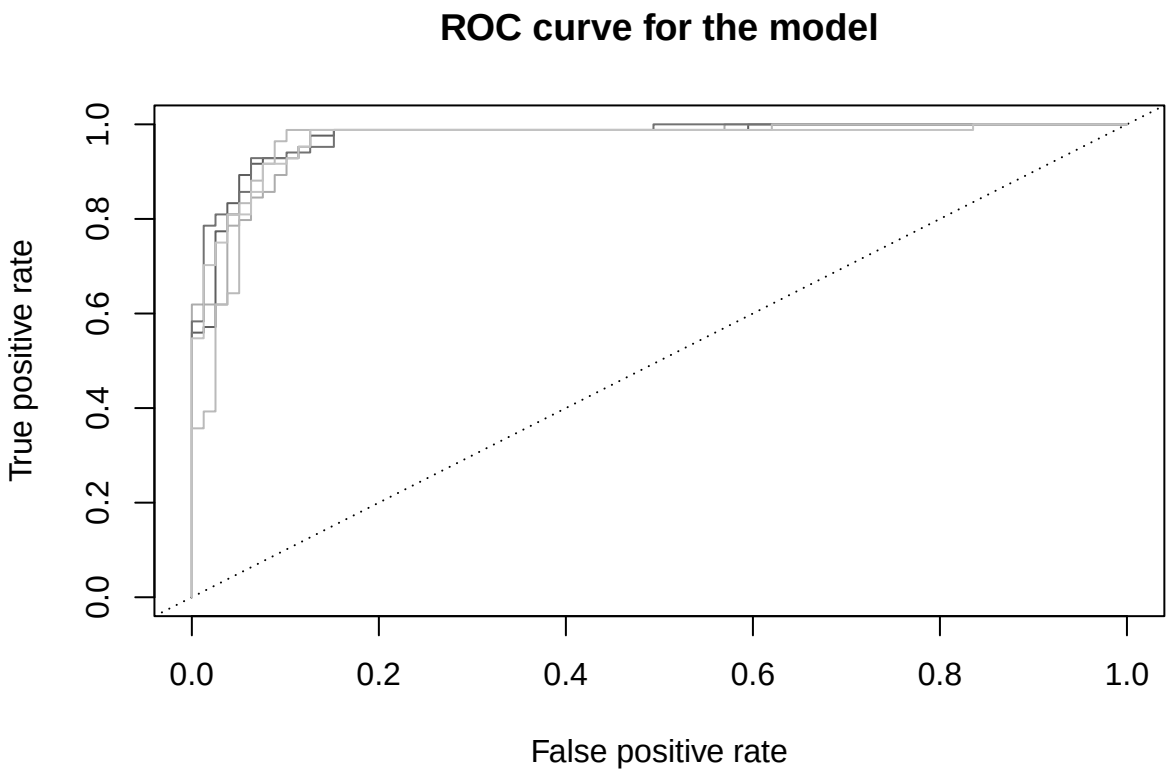
Extra analysis

```
# plot ROC Curves
plot(
    NULL,
    xlim = c(0, 1),
    ylim = c(0, 1),
    xlab = 'False positive rate',
    ylab = 'True positive rate',
    type = 'n'
)
title('ROC curve for the model')
abline(a = 0, b = 1, lty = 3)


# for each resampled CV run
eval_data <- eval_data(sc.obj.species)
for (r in 1:length(eval_data$roc.all)) {
    roc.c = eval_data$roc.all[[r]]
    lines(1 - roc.c$specificities, roc.c$sensitivities,
        col = gray(runif(1, 0.2, 0.8)))
}

# mean ROC curve
roc.summ = eval_data$roc.average[[1]]
lines(1 - roc.summ$specificities,
    roc.summ$sensitivities,
    col = 'black',
    lwd = 2)

# plot CI
x = as.numeric(rownames(roc.summ$ci))
yl = roc.summ$ci[, 1]
yu = roc.summ$ci[, 3]
polygon(1 - c(x, rev(x)), c(yl, rev(yu)), col = '#88888844' , border = NA)
```




```
#calculate from ev
tp <- sc.obj.species@eval_data[["ev"]][["tp"]] %>%
  as.data.frame() %>%
   rename_with(.cols = 1, ~"tp")

tn <- sc.obj.species@eval_data[["ev"]][["tn"]] %>%
  as.data.frame() %>%
   rename_with(.cols = 1, ~"tn")

fn <- sc.obj.species@eval_data[["ev"]][["fn"]] %>%
  as.data.frame() %>%
   rename_with(.cols = 1, ~"fn")

fp <- sc.obj.species@eval_data[["ev"]][["fp"]] %>%
  as.data.frame() %>%
   rename_with(.cols = 1, ~"fp")
```

```
sensit.specific <- bind_cols(fn, fp, tp, tn)

sensit.specific_average <- sensit.specific %>%
  summarise(across(everything(), mean, na.rm = TRUE))
```

```
## Warning: There was 1 warning in `summarise()`.
## i In argument: `across(everything(), mean, na.rm = TRUE)`.
## Caused by warning:
```

```
## ! The `...` argument of `across()` is deprecated as of dplyr 1.1.0.
## Supply arguments directly to `.fns` through an anonymous function instead.
##
##   # Previously
##   across(a:b, mean, na.rm = TRUE)
##
##   # Now
##   across(a:b, \(x) mean(x, na.rm = TRUE))
```

```
#Sensitivity = TP/(TP + FN) = (Number of true positive assessment)/(Number of all positive assessment)
(sensit.specific_average$tp)/((sensit.specific_average$tp) + (sensit.specific_average$fn))
```

```
## [1] 0.7273519
```

```
#Specificity = TN/(TN + FP) = (Number of true negative assessment)/(Number of all negative assessment)
(sensit.specific_average$tn)/((sensit.specific_average$tn) + (sensit.specific_average$fp))
```

```
## [1] 0.7417413
```

```
#Accuracy = (TN + TP)/(TN+TP+FN+FP) = (Number of correct assessments)/Number of all assessments)
(sensit.specific_average$tn) + (sensit.specific_average$tp)/
((sensit.specific_average$tn) + (sensit.specific_average$tp) + (sensit.specific_average$fn) + (sensit.sp
```

```
## [1] 58.97239
```