Model building fecal metagenomics dataset

Ece Kartal

5.12.2023

Here, we will use SIAMCAT for logistic regression model building. This is only a placeholder since metadata is not yet finalised. Based on the final set and numbers, the groups compared have to be updated via **case** variable in siamcat function below. SIAMCAT needs a feature matrix (matrix or data.frame) features (in rows) samples (in columns) metadata in a data.frame, samples as row names

```
featTable=motu.rel
metaTable=meta
# choose the meta variables to test for confounding
metatest = c("CHEMO", "IMMUN", "TARGET",  "Age", "Sex","Vital_status", "Timepoint",
             "days_to_death" , "ReadCount (Homo)", "lib_size_factor", "lib_Homo_factor",
             "Cause of death", "age_factor")
```

Overview

Function to run siamcat with centered log transformation (clr) normalization 10 fold cross validation and 10 fold re-sampling Confounder check based on fisher exact test ##################################

```
# norm:"rank.unit", "rank.std","log.std", "log.unit", "log.clr", "std", "pass"
# ml: 'lasso', 'enet', 'ridge', 'lasso_ll', 'ridge_ll', 'randomForest'
# featTable: relative abundance table
# metaTable:metadata
# label:column name of the comparison
# fileName: filenema when saving all resutls
# case: comparison e.g cancer vs control, responder vs nonresponder.
# I am not using responder since we dont have the proper metadata yet.
runsiamcat <- function(featTable, metaTable, label, fileName, case, ml, norm){
  dim(featTable)
  # create SIAMCAT object and classify
  siamcat <- siamcat(feat=featTable, meta=metaTable, label=label, case=case)

  # filter based on abundance
  siamcat <- filter.features(siamcat, filter.method = 'abundance',
                             cutoff=0.001, verbose=3)
  check.confounders(siamcat, fn.plot = paste0(PARAM$folder.output,
                             fileName, '.confounders.pdf'),
                             meta.in=metatest, verbose = 3)

  # normalize with log.clr
  siamcat <- normalize.features(siamcat, norm.method = norm,
                                feature.type = 'filtered',
                                norm.param = list(log.n0=1e-05, sd.min.q=1))
```

```
  # compute associations
  siamcat <- check.associations(siamcat, feature.type = 'normalized')

  # train model
  siamcat <- create.data.split(siamcat, num.folds = 5, num.resample = 5)
  # has to be 10 and 10, I only use 5 for testing purpuses
  siamcat <- train.model(siamcat, method = ml, verbose = 2)
  siamcat <- make.predictions(siamcat)
  siamcat <- evaluate.predictions(siamcat)
  print(siamcat@eval_data$auroc)
  # evaluation plot
  model.evaluation.plot(siamcat, fn.plot = paste0(PARAM$folder.output, Sys.Date(), '.',
                                                  fileName, '.eval.plot.pdf'))
  # interpretation plot
  model.interpretation.plot(siamcat, fn.plot = paste0(PARAM$folder.output,
                            Sys.Date(), '.', fileName,'.interpret.plot.pdf'),
                            consens.thres = 0.5,
                            heatmap.type = 'zscore')

  # save siamcat object
  save(siamcat, file = paste0(PARAM$folder.output, fileName, '.siamcat.Rdata'))
  return(siamcat)
}
```

MetaG Taxonomic Modelling

```
# prepare and subset proper meta and feat
# I subset only T1/T2 for some meaningful comparisons
metas=as.data.frame(meta) %>%
 filter(Timepoint %in% "T1/T2")
overlap_columns <- intersect(metas$ID, colnames(featTable))
metas_subset <- metas %>%
  filter(ID %in% overlap_columns)
metas_subset <- metas_subset[!duplicated(metas_subset$ID), ]
rownames(metas_subset) <- metas_subset$ID

# subset featTable
featTable <- featTable[, colnames(featTable) %in% overlap_columns]

# Check if row names are identical to column names in 'feat'
row_names_identical <- all(rownames(metas_subset) %in%
                           colnames(featTable)) && length(rownames(metas_subset)) == ncol(featTable)
# Print the result
print(row_names_identical)
```

```
## [1] TRUE
```

```
# run siamcat fpor example metavariables
# normalisation and logistic regression method has to be selected based on the data and final question,

runsiamcat(featTable, metas_subset, "Sex", "Sex", "Female", 'ridge', "log.std" )
```

```
## + starting create.label

## + removing 45 instances of NA in the label

## Label used as case:
##    Female
## Label used as control:
##    Male

## + finished create.label.from.metadata in 0.002 s

## + starting validate.data

## +++ checking overlap between labels and features

## + Removed 45 samples from the feature matrix...

## + Keeping labels of 62 sample(s).

## +++ checking sample number per class

## +++ checking overlap between samples and metadata

## + Removed 45 samples from the metadata...

## + finished validate.data in 0.208 s

## + starting filter.features

## +++ before filtering, the data have 2215 features

## +++ applying abundance filter

## +++ checking for unmapped reads

## +++ tried to remove unmapped reads but could not find any. Continue anyway.

## +++ removed 1158 features whose values did not exceed 0.001 in any sample (retaining 1057)

## +++ saving filtered features

## + finished filter.features in 0.002 s

## + starting check.confounders

## ++ metadata variables:
##  SampleName & StudienID & Filename & ID
## ++ have too many levels and have been removed from this analysis
```

```
## Warning in check.confounders(siamcat, fn.plot = paste0(PARAM$folder.output, : Some specified metadata
## Continuing with: CHEMO IMMUN TARGET Age Vital_status Timepoint days_to_death lib_size_factor lib_Hom

## ++ remove metadata variables, since all subjects have the same value
##  Timepoint

## +++ plotting conditional entropies for metadata variables

## +++ building logistic regression classifiers for metadata

## +++ plotting regression coefficients

## +++ plotting regression coefficient significance

## +++ plotting au-roc values

## +++ checking Age as a potential confounder

## ++++ continuous variable, using a Q-Q plot

## ++++ panel 1/4: Q-Q plot

## ++++ panel 2/4: X histogram

## ++++ panel 3/4: X boxplot

## ++++ panel 4/4: Y histogram

## +++ checking Vital status as a potential confounder

## ++++ discrete variable, using a bar plot

## ++++ plotting barplot

## ++++ drawing contingency table

## +++ checking Days to death as a potential confounder

## ++++ continuous variable, using a Q-Q plot

## ++++ panel 1/4: Q-Q plot

## ++++ panel 2/4: X histogram

## ++++ panel 3/4: X boxplot

## ++++ panel 4/4: Y histogram
```

```
## +++ checking Lib size factor as a potential confounder
## ++++ discrete variable, using a bar plot
## ++++ plotting barplot
## ++++ drawing contingency table
## +++ checking Lib Homo factor as a potential confounder
## ++++ discrete variable, using a bar plot
## ++++ plotting barplot
## ++++ drawing contingency table
## +++ checking Age factor as a potential confounder
## ++++ discrete variable, using a bar plot
## ++++ plotting barplot
## ++++ drawing contingency table
## +++ computing variance explained by label
## +++ computing variance explained by Age
## +++ computing variance explained by Vital_status
## +++ computing variance explained by days_to_death
## +++ computing variance explained by lib_size_factor
## +++ computing variance explained by lib_Homo_factor
## +++ computing variance explained by age_factor
## + finished check.confounders in 0.786 s
## Features normalized successfully.
## Features splitted for cross-validation successfully.
## + starting train.model
## + training ridge models on 25 training sets
```

```
## + finished train.model in 34.7 s

## Made predictions successfully.

## Evaluated predictions successfully.

## Area under the curve: 0.6534

## Plotted evaluation of predictions successfully to: /Users/ecekartal/Documents/Academics-Work/SaezLab/

## Warning in model.interpretation.select.features(feature.weights =
## feature.weights, : Restricting amount of features to be plotted to 50

## Warning in min(temp.metadata, na.rm = TRUE): no non-missing arguments to min;
## returning Inf

## Warning in max(temp.metadata, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(cur.processed.data, na.rm = TRUE): no non-missing arguments to
## max; returning -Inf

## Warning in min(temp.metadata, na.rm = TRUE): no non-missing arguments to min;
## returning Inf

## Warning in max(temp.metadata, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(cur.processed.data, na.rm = TRUE): no non-missing arguments to
## max; returning -Inf

## Warning in min(temp.metadata, na.rm = TRUE): no non-missing arguments to min;
## returning Inf

## Warning in max(temp.metadata, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(cur.processed.data, na.rm = TRUE): no non-missing arguments to
## max; returning -Inf

## Warning in min(temp.metadata, na.rm = TRUE): no non-missing arguments to min;
## returning Inf

## Warning in max(temp.metadata, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(cur.processed.data, na.rm = TRUE): no non-missing arguments to
## max; returning -Inf
```

```
## Successfully plotted model interpretation plot to: /Users/ecekartal/Documents/Academics-Work/SaezLab/
```

```
## siamcat-class object
## label()                Label object:         34 Male and 28 Female samples
## filt_feat()            Filtered features:    1057 features after abundance filtering
## associations()         Associations:         Results from association testing
##                                              with 0 significant features at alpha 0.05
## norm_feat()            Normalized features:  1057 features normalized using log.std
## data_split()           Data split:           5 cv rounds with 5 folds
## model_list()           Model list:           25 ridge models
## feature_weights()      Feature weights:      Summary of feature weights [ see also weight_matrix() ]
## pred_matrix()          Prediction matrix:    Predictions for 62 samples from 5 cv rounds
## eval_data()            Evaluation data:      Average AUC: 0.653
##
## contains phyloseq-class experiment-level object @phyloseq:
## phyloseq@otu_table()   OTU Table:            [ 2215 taxa and 62 samples ]
## phyloseq@sam_data()    Sample Data:          [ 62 samples by 37 sample variables ]
```

```
runsiamcat(featTable, metas_subset, "Vital_status", "Vital_status", "0", 'ridge', "log.std" )
```

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

```
## + removing 45 instances of NA in the label
```

```
## Label used as case:
##    0
## Label used as control:
##    1
```

```
## + finished create.label.from.metadata in 0.001 s
```

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

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

```
## + Removed 45 samples from the feature matrix...
```

```
## + Keeping labels of 62 sample(s).
```

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

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

```
## + Removed 45 samples from the metadata...
```

```
## + finished validate.data in 0.009 s
```

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

```
## +++ before filtering, the data have 2215 features
```

```
## +++ applying abundance filter

## +++ checking for unmapped reads

## +++ tried to remove unmapped reads but could not find any. Continue anyway.

## +++ removed 1158 features whose values did not exceed 0.001 in any sample (retaining 1057)

## +++ saving filtered features

## + finished filter.features in 0.001 s

## + starting check.confounders

## ++ metadata variables:
##  SampleName & StudienID & Filename & ID
## ++ have too many levels and have been removed from this analysis

## Warning in check.confounders(siamcat, fn.plot = paste0(PARAM$folder.output, : Some specified metadata
## Continuing with: CHEMO IMMUN TARGET Age Sex Timepoint days_to_death lib_size_factor lib_Homo_factor a

## ++ remove metadata variables, since all subjects have the same value
##  Timepoint

## +++ plotting conditional entropies for metadata variables

## +++ building logistic regression classifiers for metadata

## +++ plotting regression coefficients

## +++ plotting regression coefficient significance

## +++ plotting au-roc values

## +++ checking Age as a potential confounder

## ++++ continuous variable, using a Q-Q plot

## ++++ panel 1/4: Q-Q plot

## ++++ panel 2/4: X histogram

## ++++ panel 3/4: X boxplot

## ++++ panel 4/4: Y histogram

## +++ checking Sex as a potential confounder
```

```
## ++++ discrete variable, using a bar plot

## ++++ plotting barplot

## ++++ drawing contingency table

## +++ checking Days to death as a potential confounder

## ++++ continuous variable, using a Q-Q plot

## ++++ panel 1/4: Q-Q plot

## ++++ panel 2/4: X histogram

## ++++ panel 3/4: X boxplot

## ++++ panel 4/4: Y histogram

## +++ checking Lib size factor as a potential confounder

## ++++ discrete variable, using a bar plot

## ++++ plotting barplot

## ++++ drawing contingency table

## +++ checking Lib Homo factor as a potential confounder

## ++++ discrete variable, using a bar plot

## ++++ plotting barplot

## ++++ drawing contingency table

## +++ checking Age factor as a potential confounder

## ++++ discrete variable, using a bar plot

## ++++ plotting barplot

## ++++ drawing contingency table

## +++ computing variance explained by label

## +++ computing variance explained by Age

## +++ computing variance explained by Sex
```

```
## +++ computing variance explained by days_to_death

## +++ computing variance explained by lib_size_factor

## +++ computing variance explained by lib_Homo_factor

## +++ computing variance explained by age_factor

## + finished check.confounders in 0.721 s

## Features normalized successfully.

## Features splitted for cross-validation successfully.

## + starting train.model

## + training ridge models on 25 training sets

## + finished train.model in   35 s

## Made predictions successfully.

## Evaluated predictions successfully.

## Area under the curve: 0.4739

## Plotted evaluation of predictions successfully to: /Users/ecekartal/Documents/Academics-Work/SaezLab/

## Warning in model.interpretation.select.features(feature.weights =
## feature.weights, : Restricting amount of features to be plotted to 50

## Warning in min(temp.metadata, na.rm = TRUE): no non-missing arguments to min;
## returning Inf

## Warning in max(temp.metadata, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(cur.processed.data, na.rm = TRUE): no non-missing arguments to
## max; returning -Inf

## Warning in min(temp.metadata, na.rm = TRUE): no non-missing arguments to min;
## returning Inf

## Warning in max(temp.metadata, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(cur.processed.data, na.rm = TRUE): no non-missing arguments to
## max; returning -Inf
```

```
## Warning in min(temp.metadata, na.rm = TRUE): no non-missing arguments to min;
## returning Inf

## Warning in max(temp.metadata, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(cur.processed.data, na.rm = TRUE): no non-missing arguments to
## max; returning -Inf

## Warning in min(temp.metadata, na.rm = TRUE): no non-missing arguments to min;
## returning Inf

## Warning in max(temp.metadata, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(cur.processed.data, na.rm = TRUE): no non-missing arguments to
## max; returning -Inf

## Successfully plotted model interpretation plot to: /Users/ecekartal/Documents/Academics-Work/SaezLab/

## siamcat-class object
## label()                Label object:         21 1 and 41 0 samples
## filt_feat()            Filtered features:    1057 features after abundance filtering
## associations()         Associations:         Results from association testing
##                                              with 0 significant features at alpha 0.05
## norm_feat()            Normalized features:  1057 features normalized using log.std
## data_split()           Data split:           5 cv rounds with 5 folds
## model_list()           Model list:           25 ridge models
## feature_weights()      Feature weights:      Summary of feature weights [ see also weight_matrix() ]
## pred_matrix()          Prediction matrix:    Predictions for 62 samples from 5 cv rounds
## eval_data()            Evaluation data:      Average AUC: 0.474
##
## contains phyloseq-class experiment-level object @phyloseq:
## phyloseq@otu_table()   OTU Table:            [ 2215 taxa and 62 samples ]
## phyloseq@sam_data()    Sample Data:          [ 62 samples by 37 sample variables ]
```

Add confounders to metaG models

Here we can test if specific clinical variable have additional informative power to microbiome data. It has to be tested with full metadata ##

```
# load models
load(paste0(PARAM$folder.output, "Sex.siamcat.Rdata"))
siamcat.sex <- siamcat

# add confounders to model

add.meta <- function(x, n) {
  x <- add.meta.pred(x, pred.names = n, verbose = 3)
  x <- train.model(x, method = 'ridge', verbose = 2, perform.fs = TRUE)
  x <- make.predictions(x)
```

```
  x <- evaluate.predictions(x)
  return(x)
}

# combine with naive model
siamcat.vit <- add.meta(siamcat.sex, 'Vital_status')
```

```
## + starting add.meta.pred

## + starting to add metadata predictors

## +++ adding metadata predictor: Vital_status

## ++++ standardizing metadata feature Vital_status

## +++ added 1 meta-variables as predictor to the feature matrix

## + finished add.meta.pred in 0.002 s

## + starting train.model

## + training ridge models on 25 training sets

## + Performing feature selection with following parameters:

##     no_features = 100

##     method = AUC

##     direction = absolute

## + finished train.model in 19.8 s

## Made predictions successfully.

## Evaluated predictions successfully.
```

```
model.evaluation.plot('Only Sex model'= siamcat.sex,
                      'Vital status included model'= siamcat.vit,
                      fn.plot = paste0(PARAM$folder.results,  Sys.Date(),
                                       'confounders.interpret.pdf'))
```

```
## Plotted evaluation of predictions successfully to: 2024-04-15confounders.interpret.pdf
```

```
# save confounder Rdata
save(siamcat.sex, siamcat.vit,
     file = paste0(PARAM$folder.files, Sys.Date(), 'confounder.RData'))
```

Here I only showcase 2 examples, with final metadata and dataset this analysis has to be repeated btw responders and non-responders.

```r
sessionInfo()
```

```
## R version 4.3.1 (2023-06-16)
## Platform: aarch64-apple-darwin20 (64-bit)
## Running under: macOS Sonoma 14.0
##
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/lib/libRlapack.dylib;  LAPACK v
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## time zone: Europe/Berlin
## tzcode source: internal
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] readxl_1.4.3      plyr_1.8.9        ggrepel_0.9.4     SIAMCAT_2.6.0
##  [5] phyloseq_1.46.0   mlr3_0.17.0       matrixStats_1.1.0 lubridate_1.9.3
##  [9] forcats_1.0.0     stringr_1.5.1     dplyr_1.1.4       purrr_1.0.2
## [13] readr_2.1.4       tidyr_1.3.0       tibble_3.2.1      ggplot2_3.5.0
## [17] tidyverse_2.0.0
##
## loaded via a namespace (and not attached):
##   [1] beanplot_1.3.1          bitops_1.0-7            pROC_1.18.5
##   [4] gridExtra_2.3           permute_0.9-7           rlang_1.1.3
##   [7] magrittr_2.0.3          gridBase_0.4-7          ade4_1.7-22
##  [10] compiler_4.3.1          mgcv_1.9-0              vctrs_0.6.4
##  [13] reshape2_1.4.4          pkgconfig_2.0.3         shape_1.4.6
##  [16] crayon_1.5.2            fastmap_1.1.1           backports_1.4.1
##  [19] XVector_0.42.0          PRROC_1.3.1             utf8_1.2.4
##  [22] rmarkdown_2.25          tzdb_0.4.0              nloptr_2.0.3
##  [25] xfun_0.41               glmnet_4.1-8            zlibbioc_1.48.0
##  [28] mlr3misc_0.13.0         GenomeInfoDb_1.38.1     jsonlite_1.8.7
##  [31] progress_1.2.2          biomformat_1.30.0       rhdf5filters_1.14.1
##  [34] uuid_1.1-1              Rhdf5lib_1.24.0         mlr3measures_0.5.0
##  [37] prettyunits_1.2.0       parallel_4.3.1          cluster_2.1.4
##  [40] R6_2.5.1                stringi_1.8.1           RColorBrewer_1.1-3
##  [43] boot_1.3-28.1           parallelly_1.36.0       cellranger_1.1.0
##  [46] numDeriv_2016.8-1.1     Rcpp_1.0.11             iterators_1.0.14
##  [49] knitr_1.45              IRanges_2.36.0          Matrix_1.6-3
##  [52] splines_4.3.1           igraph_2.0.3            timechange_0.2.0
##  [55] tidyselect_1.2.0        rstudioapi_0.15.0       yaml_2.3.7
##  [58] mlr3tuning_0.19.1       vegan_2.6-4             codetools_0.2-19
##  [61] listenv_0.9.0           lmerTest_3.1-3          lattice_0.22-5
##  [64] Biobase_2.62.0          withr_2.5.2             evaluate_0.23
##  [67] future_1.33.0           survival_3.5-7          Biostrings_2.70.1
##  [70] infotheo_1.2.0.1        pillar_1.9.0            corrplot_0.92
##  [73] checkmate_2.3.0         foreach_1.5.2           stats4_4.3.1
##  [76] generics_0.1.3          bbotk_0.7.3             RCurl_1.98-1.13
```

```
##  [79] S4Vectors_0.40.1        hms_1.1.3               munsell_0.5.0
##  [82] scales_1.3.0            minqa_1.2.6             globals_0.16.2
##  [85] glue_1.6.2              LiblineaR_2.10-22       tools_4.3.1
##  [88] data.table_1.14.8       lme4_1.1-35.1           rhdf5_2.46.0
##  [91] grid_4.3.1              ape_5.7-1               colorspace_2.1-0
##  [94] paradox_0.11.1          nlme_3.1-163            GenomeInfoDbData_1.2.11
##  [97] palmerpenguins_0.1.1    cli_3.6.1               fansi_1.0.5
## [100] gtable_0.3.4            digest_0.6.33           BiocGenerics_0.48.1
## [103] farver_2.1.1            lgr_0.4.4               htmltools_0.5.7
## [106] multtest_2.58.0         lifecycle_1.0.4         mlr3learners_0.5.7
## [109] MASS_7.3-60
```