

```

1 #Analysis scripts for the manuscript: "Gut microbiome composition is predictive of
  incident type 2 diabetes in a population cohort of 5 572 Finnish adults" by Ruuskanen &
  Erawijantari et al.
2 #Due to sensitive health information, the data in this study are available based on a
  written application to the THL Biobank as instructed in:
  https://thl.fi/en/web/thl-biobank/for-researchers
3
4 if (!requireNamespace("BiocManager")) {
5   install.packages("BiocManager")
6 }
7
8 #Use development version of ComplexHeatmap, 2.7.11<
9 #library(devtools)
10 #install_github("jokergoo/ComplexHeatmap")
11 #devtools::install_github("slowkow/ggrepel")
12
13 packages <- c("ggplot2", "biomformat", "ggthemes", "phyloseq", "vegan", "uwot",
  "patchwork", "microbiome", "tidyverse", "reshape2", "survival", "magrittr", "ggnewscale"
  , "propr", "ComplexHeatmap", "maptree", "RColorBrewer", "rms", "viridis", "scales",
  "data.table")
14
15
16 is_installed <- function(pkg) {
17   new.pkg <- pkg[!(pkg %in% installed.packages()[, "Package"])]
18   if (length(new.pkg)) {
19     BiocManager::install(new.pkg, ask=F)
20   }
21   supply(pkg, require, character.only = TRUE)
22 }
23 is_installed(packages)
24
25 wideScreen <- function(howWide=Sys.getenv("COLUMNS")) {
26   options(width=as.integer(howWide))
27 }
28 wideScreen()
29
30 theme_set(theme_tufte(base_family = "sans", base_size = 18) + theme(panel.border =
  element_rect(colour = "black", fill = NA), axis.text = element_text(colour = "black",
  size = 18)))
31
32
33 #All data are included in the THL Biobank release package.
34 #Phenotype data is loaded from the included R object
35 load("FR_02_phenotype_data.RData")
36 #Subset to data which includes the fecal samples
37 FR02 <- FR02[!is.na(FR02$Barcode),]
38 row.names(FR02) <- FR02$Barcode
39 #Construct objects with NCBI data from SHOGUN
40 if (file.exists("microbiome_predicts_incident_T2D/ncbi_data.RDs")) {
41   ncbi_data <- readRDS("microbiome_predicts_incident_T2D/ncbi_data.RDs")
42 } else {
43   #Construct the primary phyloseq object and subset to FR02 samples.
44   ncbi_data <- biomformat::read_biom(
45     "microbiome_predicts_incident_T2D/combined_redist.species.biom") #BIOM table from the
46     SHOGUN species-level output
47   ncbi_data <- biomformat::biom_data(ncbi_data)
48   ncbi_tax_table <- strsplit(row.names(as.matrix(ncbi_data)), ";")
49   ncbi_tax_table <- matrix(unlist(ncbi_tax_table), nrow=length(ncbi_tax_table), byrow=T)
50   row.names(ncbi_data) <- row.names(ncbi_tax_table)
51   ncbi_data <- phyloseq(otu_table(as.matrix(ncbi_data), taxa_are_rows=T), tax_table(
52     ncbi_tax_table))
53   #Format the tax table.
54   colnames(tax_table(ncbi_data)) <- c("Domain", "Phylum", "Class", "Order", "Family",
55     "Genus", "Species")
56   #Combine the phenotype data with the taxa data
57   ncbi_data <- phyloseq(otu_table(ncbi_data), tax_table(ncbi_data), sample_data(FR02))
58   #remove samples with less than 50k reads (total)
59   to_be_pruned <- sample_sums(ncbi_data) > 50000
60   ncbi_data <- prune_samples(to_be_pruned, ncbi_data)

```

```

57 #remove pregnant (GRAVID==2) participants
58 ncbi_data <- subset_samples(ncbi_data, GRAVID %in% c(1, NA))
59 #remove participants who have used antibiotics in the last 6 months (BL_USE_RX_J01==1)
60 ncbi_data <- subset_samples(ncbi_data, BL_USE_RX_J01 %in% c(0, NA))
61 #remove participants with prevalent diabetes (PREVAL_DIAB==1)
62 ncbi_data <- subset_samples(ncbi_data, PREVAL_DIAB==0)
63 #remove participants with diabetes indicator values over set guidelines:
64 FR02_GLUK_NOLLA >= 7, FR02_GLUK_120 >= 11.1 & HBA1C >= 48 (ignore NA values)
65 ncbi_data <- subset_samples(ncbi_data, FR02_GLUK_NOLLA<7 | is.na(FR02_GLUK_NOLLA))
66 ncbi_data <- subset_samples(ncbi_data, FR02_GLUK_120<11.1 | is.na(FR02_GLUK_120))
67 ncbi_data <- subset_samples(ncbi_data, HBA1C<48 | is.na(HBA1C))
68 #save the final objects
69 saveRDS(ncbi_data, "microbiome_predicts_incident_T2D/ncbi_data.RDs")
70 }
71
72 #Functions
73 prediab_cat <- function(pseq){
74   data <- sample_data(pseq)
75   prediab <- ifelse(data$FR02_GLUK_NOLLA >= 5.6 & data$FR02_GLUK_NOLLA < 6.9 | data$
76     FR02_GLUK_120 >= 7.8 & data$FR02_GLUK_120 < 11 | data$HBA1C >= 39 & data$HBA1C < 47
77     , 1, 0)
78   prediab <- as.factor(prediab)
79   return(prediab)
80 }
81
82 cox_wrapper <- function(data,
83   predictors,
84   covariates,
85   status,
86   time_to_event,
87   alpha_level,
88   normalize,
89   test_ph_assumption) {
90   if(normalize) {
91     if(class(data[, predictors]) == "numeric") {
92       x <- data[, predictors]
93       data[, predictors] <- (x - mean(x, na.rm = T))/sd(x, na.rm = T)
94     } else {
95       data[, predictors] <- apply(data[, predictors], 2, FUN = function(x) {(x - mean(x
96         , na.rm = T))/sd(x, na.rm = T) })
97     }
98   }
99   ## Formulas *****
100   linear_formulas <- lapply(predictors, function(x) {
101     formula_data <- deparse(substitute(data))
102     formula <- paste0("Surv(", formula_data, "$", time_to_event, ", ", formula_data, "$",
103       status, ") ~ ", paste(covariates, collapse = "+"), " + ", x)
104     return(formula)
105   }) %>%
106   set_names(predictors)
107   ## Cox regression *****
108   print("Cox")
109   linear_cox_fit <- lapply(linear_formulas, function(x) {
110     coxph(as.formula(x), data=data, x=TRUE)
111   })
112   ## Check PH assumptions *****
113   if(test_ph_assumption) {
114     print("PH assumptions")
115     ph_assumption <- lapply(predictors, function(m) {
116       west <- cox.zph(linear_cox_fit[[m]])
117       p_values <- west$table[, "p"]
118       # significant cases
119       x <- which(p_values < 1)
120       if(length(x) == 0) {
121         return(NULL)
122       }
123       df <- data.frame(feature = m, variable_not_ph = names(x), p_value = p_values[x])
124     }) %>%

```

```

121     do.call(rbind, .) %>%
122     mutate(p_adj = p.adjust(p_value, "BH")) %>%
123     filter(p_value < alpha_level)
124   }
125   ## Results *****
126   print("Results")
127   results <- lapply(predictors, function(x) {
128     df <- summary(linear_cox_fit[[x]]$coefficients %>% as.data.frame())
129     df <- df[nrow(df), ] %>%
130     select(coef, "se(coef)", "z", "Pr(>|z|)") %>%
131     set_colnames(c("coef", "se_coef", "west_stat_value", "p")) %>%
132     mutate(west_stat = "Wald")
133     df <- df %>%
134     mutate(predictor = x)
135   }) %>%
136   do.call(rbind, .)
137   # Multiple westing correction
138   results <- results %>%
139   mutate(P_adjusted = p.adjust(p, "BH")) %>%
140   ungroup() %>%
141   group_by(predictor)
142   # Results in neat form for presentation
143   neat_results <- results %>%
144   # filter(p == min(p)) %>%
145   ungroup() %>%
146   mutate(HR = round(exp(coef), 3)) %>%
147   mutate(HR_lower_95 = round(exp(coef - 1.96*se_coef), 3),
148          HR_upper_95 = round(exp(coef + 1.96*se_coef), 3),
149          P = round(p, 5),
150          Coefficient = round(coef, 3),
151          "Coefficient SE" = round(se_coef, 3)) %>%
152   mutate(HR = paste0(HR, " (95% CI, ", HR_lower_95, "-", HR_upper_95, ")")) %>%
153   select(Predictor = predictor, Coefficient, "Coefficient SE", HR, "p", "P_adjusted",
154          "west_stat_value", "west_stat") %>%
155   mutate(HR = ifelse(is.na(Coefficient), NA, HR)) %>%
156   filter(P_adjusted < alpha_level) %>%
157   arrange(P_adjusted) %>%
158   set_colnames(c("Predictor", "Coefficient", "Coefficient SE", "HR", "P-value", "P
159                  (adjusted)", "west Statistic Value", "west Statistic"))
160   # Results in a form more convenient for further manipulations
161   results <- results %>%
162   ungroup %>%
163   mutate(PH = exp(coef)) %>%
164   mutate(p_adj = P_adjusted) %>%
165   mutate(direction = ifelse(coef < 0, "negative", "positive"))
166   if(nrow(neat_results) == 0) {
167     return(list(results = results))
168   }
169   if(test_ph_assumption) {
170     if(nrow(neat_results) == 0) {
171       return(list(results = results, ph_assumption = ph_assumption))
172     }
173     return(list(neat_results = neat_results,
174                results = results,
175                ph_assumption = ph_assumption))
176   }
177   return(list(neat_results = neat_results, results = results))
178 }
179 #preprocess data
180 if (file.exists("microbiome_predicts_incident_T2D/ncbi_data_raw_east.RDs") &&
181     file.exists("microbiome_predicts_incident_T2D/ncbi_data_raw_west.RDs") &&
182     file.exists("microbiome_predicts_incident_T2D/ncbi_data_main.RDs") && file.exists(
183     "microbiome_predicts_incident_T2D/ncbi_pca.RDs") &&
184     file.exists("microbiome_predicts_incident_T2D/ncbi_data_east.RDs") && file.exists(
185     "microbiome_predicts_incident_T2D/ncbi_pca_east.RDs") &&
186     file.exists("microbiome_predicts_incident_T2D/ncbi_data_west.RDs") && file.exists(
187     "microbiome_predicts_incident_T2D/ncbi_pca_west.RDs") &&
188     file.exists("microbiome_predicts_incident_T2D/ncbi_pca_data_east.RDs") && file.exists(

```

```

184 "microbiome_predicts_incident_T2D/ncbi_pca_data_west.RDs")) {
185   ncbi_data_raw_east <- readRDS(
186     "microbiome_predicts_incident_T2D/ncbi_data_raw_east.RDs")
187   ncbi_data_raw_west <- readRDS(
188     "microbiome_predicts_incident_T2D/ncbi_data_raw_west.RDs")
189   ncbi_data_main <- readRDS("microbiome_predicts_incident_T2D/ncbi_data_main.RDs")
190   ncbi_data_east <- readRDS("microbiome_predicts_incident_T2D/ncbi_data_east.RDs")
191   ncbi_data_west <- readRDS("microbiome_predicts_incident_T2D/ncbi_data_west.RDs")
192   ncbi_pca <- readRDS("microbiome_predicts_incident_T2D/ncbi_pca.RDs")
193   ncbi_pca_east <- readRDS("microbiome_predicts_incident_T2D/ncbi_pca_east.RDs")
194   ncbi_pca_west <- readRDS("microbiome_predicts_incident_T2D/ncbi_pca_west.RDs")
195   ncbi_pca_data_east <- readRDS(
196     "microbiome_predicts_incident_T2D/ncbi_pca_data_east.RDs")
197   ncbi_pca_data_west <- readRDS(
198     "microbiome_predicts_incident_T2D/ncbi_pca_data_west.RDs")
199 } else {
200   #Limit taxa to core in the east (EAST) set
201   core_ncbi_taxa <- core(prune_samples(meta(ncbi_data)$EAST == 1, ncbi_data) %>%
202     transform("compositional", detection = .1/100, prevalence = 1/10) %>% taxa_names())
203   ncbi_data_main <- prune_taxa(core_ncbi_taxa, ncbi_data)
204   #divide non-transformed data to east/west (EAST/WEST) sets
205   ncbi_data_raw_east <- prune_samples(meta(ncbi_data_main)$EAST == 1, ncbi_data_main)
206   ncbi_data_raw_west <- prune_samples(meta(ncbi_data_main)$EAST == 0, ncbi_data_main)
207   #CLR-transform raw counts
208   ncbi_data_main <- transform(ncbi_data_main, "clr")
209   #calculate additional variables
210   PREDIAB <- prediab_cat(ncbi_data_main)
211   NON_HDL <- sample_data(ncbi_data)$KOL - sample_data(ncbi_data)$HDL
212   #calculate diversity
213   ncbi_diversity <- estimate_richness(ncbi_data, measures = c("Observed", "Shannon"))
214   #reduce metadata to useful columns
215   useful_variables <- c("BL_AGE", "BMI", "MEN", "SYSTEM", "CURR_SMOKE", "TRIG",
216     "INCIDENT_DIAB_T2", "DIAB_T2_AGEDIFF", "EAST")
217   sample_data(ncbi_data_main) <- sample_data(ncbi_data_main)[,sample_variables(
218     ncbi_data_main) %in% useful_variables]
219   #separate transformed and curated data to east/west (EAST/WEST) sets
220   ncbi_data_east <- prune_samples(meta(ncbi_data_main)$EAST == 1, ncbi_data_main)
221   ncbi_data_west <- prune_samples(meta(ncbi_data_main)$EAST == 0, ncbi_data_main)
222   #calculate 10 first PCAs with full community
223   ncbi_data_raw_clr <- transform(ncbi_data, "clr")
224   ncbi_pca <- ordinate(ncbi_data_raw_clr, "RDA")
225   ncbi_pca_data <- as.data.frame(scores(ncbi_pca, choices = c(1:10))$sites)
226   ncbi_pca_east <- ordinate(prune_samples(meta(ncbi_data_raw_clr)$EAST == 1,
227     ncbi_data_raw_clr), "RDA")
228   ncbi_pca_data_east <- as.data.frame(scores(ncbi_pca_east, choices = c(1:10))$sites)
229   ncbi_pca_west <- ordinate(prune_samples(meta(ncbi_data_raw_clr)$EAST == 0,
230     ncbi_data_raw_clr), "RDA")
231   ncbi_pca_data_west <- as.data.frame(scores(ncbi_pca_west, choices = c(1:10))$sites)
232   #combine with additional data
233   sample_data(ncbi_data_main) <- cbind(sample_data(ncbi_data_main), PREDIAB, NON_HDL,
234     ncbi_diversity, ncbi_pca_data)
235   sample_data(ncbi_data_east) <- cbind(sample_data(ncbi_data_east), PREDIAB = PREDIAB[
236     which(meta(ncbi_data_main)$EAST == 1)], NON_HDL = NON_HDL[which(meta(ncbi_data_main)$
237     EAST == 1)], ncbi_diversity[which(meta(ncbi_data_main)$EAST == 1)],,
238     ncbi_pca_data_east)
239   sample_data(ncbi_data_west) <- cbind(sample_data(ncbi_data_west), PREDIAB = PREDIAB[
240     which(meta(ncbi_data_main)$EAST == 0)], NON_HDL = NON_HDL[which(meta(ncbi_data_main)$
241     EAST == 0)], ncbi_diversity[which(meta(ncbi_data_main)$EAST == 0)],,
242     ncbi_pca_data_west)
243   saveRDS(ncbi_data_raw_east, "microbiome_predicts_incident_T2D/ncbi_data_raw_east.RDs")
244   saveRDS(ncbi_data_raw_west, "microbiome_predicts_incident_T2D/ncbi_data_raw_west.RDs")
245   saveRDS(ncbi_data_main, "microbiome_predicts_incident_T2D/ncbi_data_main.RDs")
246   saveRDS(ncbi_data_east, "microbiome_predicts_incident_T2D/ncbi_data_east.RDs")
247   saveRDS(ncbi_data_west, "microbiome_predicts_incident_T2D/ncbi_data_west.RDs")
248   saveRDS(ncbi_pca, "microbiome_predicts_incident_T2D/ncbi_pca.RDs")
249   saveRDS(ncbi_pca_east, "microbiome_predicts_incident_T2D/ncbi_pca_east.RDs")
250   saveRDS(ncbi_pca_west, "microbiome_predicts_incident_T2D/ncbi_pca_west.RDs")
251   saveRDS(ncbi_pca_data_east, "microbiome_predicts_incident_T2D/ncbi_pca_data_east.RDs")
252   saveRDS(ncbi_pca_data_west, "microbiome_predicts_incident_T2D/ncbi_pca_data_west.RDs")

```

```

236 }
237
238 #Filter features based on corrected p-values in the east dataset
239 #set variables
240 alpha_level <- 0.05 #to filter
241 status <- "INCIDENT_DIAB_T2"
242 time_to_event <- "DIAB_T2_AGEDIFF"
243 ncbi_cox_data_east <- cbind(meta(ncbi_data_east), as.matrix(t(otu_table(ncbi_data_east
244 )))
245 predictors <- c("Shannon", "Observed", colnames(ncbi_pca_data_east), taxa_names(
246 ncbi_data_east))
247 covariates <- c("BL_AGE", "BMI", "MEN", "SYSTEM", "NON_HDL", "CURR_SMOKE", "TRIG")
248 splines <- TRUE
249 normalize <- TRUE
250 test_ph_assumption <- FALSE
251 #Cox regression with previously defined function
252 set.seed(11235)
253 ncbi_cox_east <- cox_wrapper(data = ncbi_cox_data_east,
254                             predictors = predictors,
255                             covariates = covariates,
256                             alpha_level = alpha_level,
257                             status = status,
258                             time_to_event = time_to_event,
259                             normalize = normalize,
260                             test_ph_assumption = test_ph_assumption)
261
262 ncbi_cox_results_east <- merge(ncbi_cox_east$neat_results, as.data.frame(ncbi_data_east@
263 tax_table@.Data), by.x="Predictor", by.y="row.names")
264 ncbi_cox_results_east <- ncbi_cox_results_east[order(-ncbi_cox_results_east$Coefficient)
265 ,]
266 ncbi_cox_results_east$Species <- gsub("s_", "", ncbi_cox_results_east$Species)
267 ncbi_cox_results_east$Species <- gsub("_", " ", ncbi_cox_results_east$Species)
268 ncbi_cox_results_east$Family <- gsub("f_", "", ncbi_cox_results_east$Family)
269
270 #Correlations and clustering between the associated taxa in east data
271 otu_table_assoc_taxa <- as.data.frame(otu_table(prune_taxa(ncbi_cox_east$neat_results$
272 Predictor, ncbi_data_raw_east)))
273 rownames(otu_table_assoc_taxa) <- ncbi_cox_results_east$Species[match(rownames(
274 otu_table_assoc_taxa), ncbi_cox_results_east$Predictor)]
275 set.seed(11235)
276 proprmatrix <- propr(t(otu_table_assoc_taxa), metric = "rho", p = 100)
277 clusters_assoc <- hclust(dist(proprmatrix@matrix), method = "ward.D2")
278 #Compute the Kelley-Gardner-Sutcliffe penalty function for a hierarchical cluster tree,
279 to determine optimal number of clusters
280 op_k <- kgs(clusters_assoc, dist(proprmatrix@matrix), maxclus = 20)
281 op_k <- as.numeric(names(op_k[which(op_k == min(op_k))]))
282 cluster_ids <- cutree(tree = clusters_assoc, k = op_k)
283 svg("microbiome_predictions_incident_T2D/clusters.svg", width=10, height=10)
284 plot(clusters_assoc)
285 rect.hclust(clusters_assoc, k = op_k, border = 2:7)
286 dev.off()
287
288 heatmap_annotation <- data.frame(Species = rownames(proprmatrix@matrix), Cluster =
289 cluster_ids)
290 heatmap_annotation$Predictor <- ncbi_cox_results_east$Predictor[match(heatmap_annotation
291 $Species, ncbi_cox_results_east$Species)]
292
293 #Clustering correlating significant taxa for east and west data
294 #Combine read counts of clusters and calculate their CLR values
295 taxa_clusters <- merge(heatmap_annotation[c("Cluster")], ncbi_cox_results_east[c(
296 "Species", "Predictor")], by.x = "row.names", by.y = "Species")
297 taxa_clusters$Cluster <- factor(taxa_clusters$Cluster, levels = 1:length(unique(
298 taxa_clusters$Cluster)))
299
300 cluster_phylo_east <- ncbi_data_raw_east
301 cluster_phylo_west <- ncbi_data_raw_west
302 index_taxa <- c()
303 for (cluster in levels(taxa_clusters$Cluster)) {
304   taxa_to_merge <- taxa_clusters$Predictor[which(taxa_clusters$Cluster == cluster)]

```

```

294 cluster_phylo_east <- merge_taxa(cluster_phylo_east, taxa_to_merge, archetype=1)
295 cluster_phylo_west <- merge_taxa(cluster_phylo_west, taxa_to_merge, archetype=1)
296 index_taxa[cluster] <- taxa_to_merge[1]
297 }
298 cluster_phylo_east <- transform(cluster_phylo_east, "clr")
299 cluster_phylo_west <- transform(cluster_phylo_west, "clr")
300 #Retain only clusters
301 cluster_phylo_east <- prune_taxa(index_taxa, cluster_phylo_east)
302 cluster_phylo_west <- prune_taxa(index_taxa, cluster_phylo_west)
303 taxa_names(cluster_phylo_east) <- paste0("Cluster_", taxa_clusters$Cluster[match(
taxa_names(cluster_phylo_east), taxa_clusters$Predictor)])
304 taxa_names(cluster_phylo_west) <- paste0("Cluster_", taxa_clusters$Cluster[match(
taxa_names(cluster_phylo_west), taxa_clusters$Predictor)])
305
306 #test the individual taxa and clusters in the east data
307 #set variables
308 alpha_level <- 1 #to include everything in the results
309 status <- "INCIDENT_DIAB_T2"
310 time_to_event <- "DIAB_T2_AGEDIFF"
311 ncbi_cox_data_east_2 <- cbind(meta(ncbi_data_east), as.matrix(t(otu_table(ncbi_data_east
))), as.matrix(t(otu_table(cluster_phylo_east))))
312 predictors <- c(ncbi_cox_results_east$Predictor, taxa_names(cluster_phylo_east), "PC1")
313 covariates <- c("BL_AGE", "BMI", "MEN", "SYSTEM", "NON_HDL", "CURR_SMOKE", "TRIG")
314 splines <- TRUE
315 normalize <- TRUE
316 test_ph_assumption <- FALSE
317 #Cox regression with previously defined function
318 set.seed(11235)
319 ncbi_cox_east_2 <- cox_wrapper(data = ncbi_cox_data_east_2,
320                               predictors = predictors,
321                               covariates = covariates,
322                               alpha_level = alpha_level,
323                               status = status,
324                               time_to_event = time_to_event,
325                               normalize = normalize,
326                               test_ph_assumption = test_ph_assumption)
327
328 ncbi_cox_results_east_2 <- data.frame(ncbi_cox_east_2$neat_results)
329 ncbi_cox_results_east_2 <- merge(ncbi_cox_results_east_2[c("Predictor", "Coefficient",
"HR", "P.value")], ncbi_cox_results_east[c("Predictor", "Family", "Species")], by =
"Predictor", all = TRUE)
330 ncbi_cox_results_east_2 <- ncbi_cox_results_east_2[order(-ncbi_cox_results_east_2$
Coefficient),]
331 ncbi_cox_results_east_2$Set <- "East"
332
333 #test the individual taxa and clusters in the west data
334 #use same variables as for previous model run (thus not repeated here)
335 ncbi_cox_data_west <- cbind(meta(ncbi_data_west), as.matrix(t(otu_table(ncbi_data_west
))), as.matrix(t(otu_table(cluster_phylo_west))))
336 #Cox regression with previously defined function
337 set.seed(11235)
338 ncbi_cox_west <- cox_wrapper(data = ncbi_cox_data_west,
339                               predictors = predictors,
340                               covariates = covariates,
341                               alpha_level = alpha_level,
342                               status = status,
343                               time_to_event = time_to_event,
344                               normalize = normalize,
345                               test_ph_assumption = test_ph_assumption)
346
347 ncbi_cox_results_west <- data.frame(ncbi_cox_west$neat_results)
348 ncbi_cox_results_west <- merge(ncbi_cox_results_west[c("Predictor", "Coefficient", "HR"
, "P.value")], ncbi_cox_results_east[c("Predictor", "Family", "Species")], by =
"Predictor", all = TRUE)
349 ncbi_cox_results_west <- ncbi_cox_results_west[order(-ncbi_cox_results_west$Coefficient
),]
350 ncbi_cox_results_west$Set <- "West"
351
352 #save results

```



```

353 results_out_east <- rbind(data.frame(ncbi_cox_east$neat_results), data.frame(
ncbi_cox_east_2$neat_results[which(grepl("Cluster", ncbi_cox_east_2$neat_results$
Predictor)),]))
354 results_out_west <- data.frame(ncbi_cox_west$neat_results)
355 results_out_east <- merge(results_out_east[c("Predictor", "Coefficient", "HR", "P.value"
, "P..adjusted."),], as.data.frame(ncbi_data_east@tax_table@.Data)["Species"], by.x=
"Predictor", by.y="row.names", all.x = TRUE)
356 results_out_west <- merge(results_out_west[c("Predictor", "Coefficient", "HR", "P.value"
, "P..adjusted."),], as.data.frame(ncbi_data_west@tax_table@.Data)["Species"], by.x=
"Predictor", by.y="row.names", all.x = TRUE)
357 results_out_west$P..adjusted. <- NA
358 results_out_east[which(grepl("Cluster", results_out_east$Predictor)),]$P..adjusted. <- NA
359 results_out <- merge(results_out_east, results_out_west, by="Predictor", suffixes=c(
".east", ".west"))
360 result_order <- results_out[rev(order(results_out$Coefficient.east)),]$Predictor
361 result_order <- c("PC1", paste0("Cluster_", 1:5), result_order[which(grepl("sp",
result_order))])
362 results_out <- results_out[match(result_order, results_out$Predictor),]
363 results_out[~which(is.na(results_out$Species.east)), "Predictor"] <- as.character(
results_out[~which(is.na(results_out$Species.east)), "Species.east"])
364 results_out$Predictor <- gsub("s_", "", results_out$Predictor)
365 results_out$Predictor <- gsub(" ", " ", results_out$Predictor)
366 names(results_out) <- gsub("\\.\\.", ".", names(results_out))
367 results_out <- results_out[,!names(results_out) %in% c("Species.east", "Species.west",
"P.adjusted.west")]
368 results_out[c("P.value.east", "P.adjusted.east", "P.value.west")] <- lapply(results_out[
c("P.value.east", "P.adjusted.east", "P.value.west")], function(x) round(x, 4))
369 write.csv(results_out, "microbiome_predicts_incident_T2D/Table_S1.csv", row.names=F)
370
371 #plot heatmap of taxa associations, clustering, and hazard ratios in the east data
372 newnames <- lapply(rownames(proprmatrix@matrix), function(x) bquote(italic(.x))))
373 heatmap_annotation$HR <- gsub("([0-9]\\.[0-9]*)[[:space:]].*", "\\1",
ncbi_cox_results_east$HR[match(heatmap_annotation$Predictor, ncbi_cox_results_east$
Predictor)])
374 heatmap_annotation$HR <- round(as.numeric(as.character(heatmap_annotation$HR)), 1)
375 heatmap_annotation$HR <- factor(heatmap_annotation$HR, levels = rev(seq(0.8, 1.2, 0.1)))
376
377 ann_colors <- list(HR = brewer.pal(n = 5, name = "BrBG"), Cluster = brewer.pal(n = 10,
name = "Paired")[-seq(1,9,2)])
378 names(ann_colors$HR) <- levels(heatmap_annotation$HR)
379 names(ann_colors$Cluster) <- c("1", "2", "3", "4", "5")
380 ann_colors$Cluster <- factor(ann_colors$Cluster, levels = ann_colors$Cluster[c(4,2,5,6,3
,1)])
381 heatmap_colors <- rev(brewer.pal(n = 10, name = "RdBu"))
382 heatmap_colors[c(5,6)] <- "#FFFFFF"
383 svg("microbiome_predicts_incident_T2D/correlations.svg", width=15, height=15)
384 pheatmap(proprmatrix@matrix, labels_row = as.expression(newnames), labels_col =
as.expression(newnames), annotation_row = heatmap_annotation[4], treeheight_row = 0,
annotation_col = heatmap_annotation[2], annotation_colors = ann_colors, cutree_rows =
op_k, cutree_cols = op_k, clustering_method = "ward.D2", color = heatmap_colors, breaks
= seq(-1, 1, length.out = 11), legend_breaks = seq(-1, 1, length.out = 11), cellwidth=10
, cellheight=10)
385 dev.off()
386
387 #plot HR of both west and east data
388 ncbi_cox_results <- rbind(ncbi_cox_results_east_2, ncbi_cox_results_west)
389
390 Species <- c()
391 Family <- c()
392 Set <- c()
393 Facet <- c()
394 HR <- c()
395 HR1 <- c()
396 HR2 <- c()
397
398 for (i in 1:length(ncbi_cox_results$Predictor)){
399   Species[[i]] <- ifelse(is.na(ncbi_cox_results$Species[i]), sub("_", " ",
ncbi_cox_results$Predictor[i]), as.character(ncbi_cox_results$Species[i]))
400   Family[[i]] <- ifelse(is.na(ncbi_cox_results$Family[i]), NA, as.character(

```

```

ncbi_cox_results$Family[i]))
401 HR[[i]] <- str_split(ncbi_cox_results$HR[i], " ")[[1]][1]
402 HR_range <- str_split(ncbi_cox_results$HR[i], " ")[[1]][4]
403 HR1[[i]] <- str_split(HR_range, "-")[1][1]
404 HR2_bef <- str_split(HR_range, "-")[1][2]
405 HR2[[i]] <- substr(HR2_bef, 1, nchar(HR2_bef)-1)
406 Set[[i]] <- ncbi_cox_results$Set[i]
407 Facet[[i]] <- ifelse(is.na(ncbi_cox_results$Family[i]), "Grouping", "Taxa")
408 HRdf <- data.frame(Species = Species,
409                   Family = Family,
410                   Set = Set,
411                   Facet = Facet,
412                   HR = HR,
413                   HR1 = HR1,
414                   HR2 = HR2)
415 }
416
417 family_color_map <- data.frame(Color = c("#7b562e", "#9bb940", "#c5bb9a", "darkred",
418 "#ff4ae3", "#339a00", "#d78343", "#5f96d6", "black"),
419 Family = c("Bacteroidaceae", "Clostridiaceae", "Eggerthellaceae", "Eubacteriaceae",
420 "Lachnospiraceae", "Oscillospiraceae", "Rickenellaceae", "Sutterellaceae", NA))
421
422 HRdf$Species <- factor(HRdf$Species, levels = c(paste0("Cluster ", 5:1), "PC1",
423 as.character(HRdf[which(HRdf$Set %in% "East" & HRdf$Facet %in% "Taxa"),][order(HRdf[
424 which(HRdf$Set %in% "East" & HRdf$Facet %in% "Taxa"),]$HR),]$Species))) #order features
425 by effect size in the east data
426
427 p <- ggplot(data = HRdf, aes(y = Species, x = as.numeric(as.character(HR)), color =
428 Family)) +
429   geom_pointrange(aes(xmin=as.numeric(as.character(HR1)), xmax=as.numeric(
430 as.character(HR2))), lwd = 1) +
431   scale_x_continuous(limits = c(0.6, 1.45)) +
432   scale_color_manual(name = "Family", values = as.character(family_color_map$Color))
433   +
434   guides(color = guide_legend(override.aes = list(size = 1.4))) +
435   xlab("HR") + ylab("Species") +
436   geom_vline(xintercept=c(1.0), linetype="dotted") +
437   theme(axis.text.y = element_text(face = "italic"), legend.text = element_text(face
438 = "italic"), axis.title.y = element_blank()) +
439   facet_grid(Facet~Set, scales = "free")
440
441 ggsave("microbiome_predicts_incident_T2D/HR_comparison.svg", plot=p, units="in", width=
442 15, height=10)
443
444 #Plot Kaplan-Meier curves
445 kp_predictors <- ncbi_cox_results$Predictor[which(ncbi_cox_results$P.value <
446 0.05)]
447 kp_covariates <- covariates
448 kp_time_to_event <- time_to_event
449 kp_status <- status
450 kp_data <- ncbi_cox_data$west[,which(colnames(ncbi_cox_data$west) %in% c(kp_status,
451 kp_time_to_event, kp_predictors, kp_covariates))]
452 kp_time <- seq(0, max(kp_data$DIAB_T2_AGEDIFF), by = .01)
453 kp_list <- list(NULL)
454 for (time in 1:length(kp_time)) {
455   kp_table <- lapply(kp_predictors, function(x) {
456     return_table <- data.frame(groupkm(kp_data[x], Surv(kp_data$DIAB_T2_AGEDIFF, kp_data
457 $INCIDENT_DIAB_T2), g=4, u=kp_time[time], pl=FALSE))
458     return_table$Predictor <- x
459     return_table$quantile <- c(1:4)
460     return(return_table)
461   })
462   kp_table <- do.call(rbind, kp_table)
463   kp_table$time <- kp_time[time]
464   kp_list[[time]] <- kp_table
465 }
466
467 kp_list <- do.call(rbind, kp_list)
468 kp_predictors <- recode(kp_predictors, 'sp2673' = "[Clostridium] citroniae", 'sp2671' =
469 "[Clostridium] bolteae", 'sp2697' = "Tyzzerella nexilis", 'sp2638' = "[Ruminococcus]

```



```

gnavus")
455 kp_predictors <- gsub("_", " ", kp_predictors)
456 kp_list$Predictor <- recode(kp_list$Predictor, 'sp2673' = "[Clostridium] citroniae",
'sp2671' = "[Clostridium] bolteaе", 'sp2697' = "Tyzzzerella nexilis", 'sp2638' =
"[Ruminococcus] gnavus")
457 kp_list$Predictor <- gsub("_", " ", kp_list$Predictor)
458 kp_list$Predictor <- factor(kp_list$Predictor, levels = kp_predictors)
459
460 p <- ggplot(data = kp_list, aes(y = KM, x = time, group = quantile)) +
461   geom_line(aes(color = quantile)) +
462   scale_color_viridis(labels = c("Min to Q1", "Q1 to Q2", "Q2 to Q3", "Q3 to max")) +
463   scale_y_continuous(breaks = pretty_breaks()) +
464   guides(color = guide_legend(override.aes = list(size = 1.4)), fill = "none") +
465   xlab("Time (years)") + ylab("Survival without type 2 diabetes") +
466   labs(color = "Relative\nabundance\nrange") +
467   facet_wrap(~ Predictor)
468 ggsave("microbiome_predicts_incident_T2D/KP_plot.svg", plot=p, units="cm", width=30,
height=20)
469
470 #Plot distributions of the quartiles (for inlays in the KP-plot)
471 quartile_data <- lapply(kp_data[ncbi_cox_results_west$Predictor[which(
ncbi_cox_results_west$P.value < 0.05)]],
472   function(x) data.frame(x_value = density(x)$x,
473     y_value = density(x)$y,
474     quartile = factor(paste0("Q", findInterval(density(x)$x,
quantile(x, prob=c(0, 0.25, 0.5, 0.75, 1)), all.inside=T))))))
475 quartile_data <- data.frame(rbindlist(quartile_data, idcol="Predictor"))
476 quartile_data$Predictor <- recode(quartile_data$Predictor, 'sp2673' = "[Clostridium]
citroniae", 'sp2671' = "[Clostridium] bolteaе", 'sp2697' = "Tyzzzerella nexilis",
'sp2638' = "[Ruminococcus] gnavus")
477 quartile_data$Predictor <- gsub("_", " ", quartile_data$Predictor)
478 quartile_data$Predictor <- factor(quartile_data$Predictor, levels = kp_predictors)
479
480 p <- ggplot(quartile_data, aes(x_value, y_value)) +
481   geom_line() +
482   geom_ribbon(aes(ymin=0, ymax=y_value, fill=quartile)) +
483   scale_fill_viridis(labels = c("Q1", "Q2", "Q3", "Q4"), discrete=T) +
484   guides(fill = "none") +
485   theme(axis.title = element_blank()) +
486   facet_wrap(~ Predictor)
487 ggsave("microbiome_predicts_incident_T2D/KP_plot_quartiles.svg", plot=p, units="cm",
width=30, height=20)
488
489 #Format and print Table 1
490 table_data <- data.table(meta(ncbi_data))
491 table_data$NON_HDL <- table_data$KOL - table_data$HDL
492 table_variables <- c("BL_AGE", "BMI", "SYSTM", "NON_HDL", "FR02_GLUK_NOLLA",
"FR02_GLUK_120", "HBA1C", "TRIG")
493 table_variables2 <- c("CURR_SMOKE", "MEN", "EAST")
494 table_data <- table_data[, .SD, .SDcols = c(table_variables, table_variables2,
"INCIDENT_DIAB_T2")]
495 table_data$MEN <- ifelse(table_data$MEN == 0, 1, 0) #invert the variable to count
female (not male) participants as 1's for the table
496 table1 <- transpose(merge(rbind(table_data[, c(INCIDENT_DIAB_T2 = "all", lapply(.SD,
function(x) paste0(round(mean(x, na.rm=T),1), "+-", round(sd(x, na.rm=T),1))) ), .SDcols
=table_variables],
497   table_data[, lapply(.SD, function(x) paste0(round(mean(x, na.rm=T),1), "+-", round(
sd(x, na.rm=T),1))), by=INCIDENT_DIAB_T2, .SDcols=table_variables]),
498   rbind(table_data[, c(INCIDENT_DIAB_T2 = "all", lapply(.SD, function(x) paste0(table(
x)[2], " (", round(table(x)[2]/length(x)*100, 1), ")"))), .SDcols=table_variables2],
499   table_data[, lapply(.SD, function(x) paste0(table(x)[2], " (", round(table(x)[2]/
length(x)*100, 1), ")"))], by=INCIDENT_DIAB_T2, .SDcols=table_variables2], by =
"INCIDENT_DIAB_T2", keep.names = "col", make.names = "INCIDENT_DIAB_T2")
500 table1 <- table1[,c("col", "all", "1", "0")]
501 p_values_cont <- as.vector(as.matrix(table_data[, lapply(.SD, function(x) signif(
wilcox.test(x ~ INCIDENT_DIAB_T2)$p.value, digits = 2)), .SDcols = table_variables]))
502 for (i in 1:length(table_variables2)) {
503   p_values_cont[length(table_variables)+i] <- signif(fisher.test(rbind(table(
table_data[table_data$INCIDENT_DIAB_T2 == 1,][[table_variables2[i]]], table(

```

```

    table_data[table_data$INCIDENT_DIAB_T2 == 0, ][[table_variables2[i]]]))$p.value,
    digits = 2)
504 }
505 table1$p_value <- p_values_cont
506 table1[col == "MEN", col := "WOMEN"] #change the name of the "MEN" variable to "WOMEN"
to reflect the inversion
507 table1 <- rbind(list("N", nrow(table_data), nrow(table_data[table_data$INCIDENT_DIAB_T2
== 1,])), nrow(table_data[table_data$INCIDENT_DIAB_T2 == 0,])), table1, fill=T)
508 colnames(table1) <- c("Variable", "Total", "With Incident T2D", "Without Incident T2D",
"P-value")
509
510 table_variables3 <- c("CURR_SMOKE", "MEN", "INCIDENT_DIAB_T2")
511 table2 <- transpose(merge(table_data[, lapply(.SD, function(x) paste0(round(mean(x,
na.rm=T),1), "+-", round(sd(x, na.rm=T),1))), by=EAST, .SDcols=table_variables],
512 table_data[, lapply(.SD, function(x) paste0(table(x)[2], " (", round(table(x)[2]/
length(x)*100, 1), ")")), by=EAST, .SDcols=table_variables3], by = "EAST"),
keep.names = "col", make.names = "EAST")
513 table2 <- table2[,c("col", "1", "0")]
514 p_values_cont2 <- as.vector(as.matrix(table_data[, lapply(.SD, function(x) signif(
wilcox.test(x ~ EAST)$p.value, digits = 2)), .SDcols = table_variables]))
515 for (i in 1:length(table_variables3)) {
516 p_values_cont2[length(table_variables)+i] <- signif(fisher.test(rbind(table(
table_data[table_data$EAST == 1, ][[table_variables3[i]]], table(table_data[
table_data$EAST == 0, ][[table_variables3[i]]]))$p.value, digits = 2)
517 }
518 table2$p_value <- p_values_cont2
519 table2[col == "MEN", col := "WOMEN"] #change the name of the "MEN" variable to "WOMEN"
to reflect the inversion
520 table2 <- rbind(list("N", nrow(table_data[table_data$EAST == 1,])), nrow(table_data[
table_data$EAST == 0,])), table2, fill=T)
521 colnames(table2) <- c("Variable", "From Eastern Finland", "From Western Finland",
"P-value")
522 table_out <- merge(table1, table2, by = "Variable", all = TRUE)
523 table_out <- table_out[,c("N", "WOMEN", "EAST", "INCIDENT_DIAB_T2", table_variables,
"CURR_SMOKE"),]
524 write.csv(table_out, "microbiome_predicts_incident_T2D/Table1.csv", row.names=F)
525
526 #Correlations and clustering between the associated taxa in west data
527 otu_table_assoc_taxa_west <- as.data.frame(otu_table(prune_taxa(ncbi_cox_west$
neat_results$Predictor, ncbi_data_raw_west)))
528 rownames(otu_table_assoc_taxa_west) <- ncbi_cox_results_west$Species[match(rownames(
otu_table_assoc_taxa_west), ncbi_cox_results_west$Predictor)]
529 set.seed(11235)
530 proprmatrix_west <- propr(t(otu_table_assoc_taxa_west), metric = "rho", p = 100)
531 clusters_assoc_west <- hclust(dist(proprmatrix_west@matrix), method = "ward.D2")
532 #Compute the Kelley-Gardner-Sutcliffe penalty function for a hierarchical cluster tree,
to determine optimal number of clusters
533 op_k_west <- kgs(clusters_assoc_west, dist(proprmatrix_west@matrix), maxclus = 20)
534 op_k_west <- as.numeric(names(op_k_west[which(op_k_west == min(op_k_west))]))
535 cluster_ids_west <- cutree(tree = clusters_assoc_west, k = op_k_west)
536 svg("microbiome_predicts_incident_T2D/clusters_west.svg", width=10, height=10)
537 plot(clusters_assoc_west)
538 rect.hclust(clusters_assoc_west, k = op_k_west, border = 2:7)
539 dev.off()
540
541 #plot heatmap of taxa associations, clustering, and hazard ratios in the west data
542 newnames_west <- lapply(rownames(proprmatrix_west@matrix), function(x) bquote(italic(. (x
))))
543
544 #clusters are identical in membership of taxa in the same cluster, so we can just copy
the cluster annotation from east data to west data to keep cluster colors and naming
consistent
545 heatmap_annotation_west <- heatmap_annotation
546 #get correct hazard ratios for west data
547 heatmap_annotation_west$HR <- gsub("([0-9]\\.[0-9]*)[:,space:]].*", "\\1",
ncbi_cox_results_west$HR[match(heatmap_annotation_west $Predictor, ncbi_cox_results_west
$Predictor)])
548 heatmap_annotation_west$HR <- round(as.numeric(as.character(heatmap_annotation_west $HR
)), 1)

```

```
549 heatmap_annotation_west$HR <- factor(heatmap_annotation_west $HR, levels = rev(seq(0.8,
550 1.2, 0.1)))
551 svg("microbiome_predicts_incident_T2D/correlations_west.svg", width=15, height=15)
552 pheatmap(proprmatrix_west@matrix, labels_row = as.expression(newnames_west), labels_col
= as.expression(newnames_west), annotation_row = heatmap_annotation_west[4],
treeheight_row = 0, annotation_col = heatmap_annotation_west[2], annotation_colors =
ann_colors, cutree_rows = op_k_west, cutree_cols = op_k_west, clustering_method =
"ward.D2", color = heatmap_colors, breaks = seq(-1, 1, length.out = 11), legend_breaks =
seq(-1, 1, length.out = 11), cellwidth=10, cellheight=10)
553 dev.off()
554
555 save.image("microbiome_predicts_incident_T2D/Analysis.RData")
```