Supplementary 1 MATERIALS AND METHODS

KEY RESOURCES TABLE

RESOURCE	SOURCE	IDENTIFIER / WEBSITE
Nasopharyngeal swab		
Nasopharyngeal swab	COPAN	COPAN Diagnostics 56380CS01
Transport medium UTM tubes	COPAN	COPAN Diagnostics 3C075N
DNA extraction		
Sputolysin	Calbiochem	CAS 578517-Calbiochem
QIAamp UCP Pathogen Mini Kit	QIAGEN	Cat. No. / ID: 50214
Pathogen Lysis Tubes L	QIAGEN	Cat. No. / ID: 19092
16S rRNA Amplicon PCR		
Phusion Plus PCR Master Mix	Thermo Fisher Scientific	Cat. No. F631S
515f: 5´-GTGCCAGCMGCCGCGGTAA-3´	López-Filoy et al., 2022 [23]	doi:10.3389/fcimb.2022.884272
806r: 5´-GGACTACHVGGGTWTCTAAT-3´	López-Filoy et al., 2022 [23]	doi:10.3389/fcimb.2022.884272
DNA Clean &Concentrator	Zymo	Cat. No. D4013
Sequencing		
Nextera XT Library Preparation Kit	Illumina	FC-131-1096
Deposited data		
V4 region of 16S rRNA sequencing data	This study	https://www.ncbi.nlm.nih.gov/bioproject/PRJNA981220/
V3-V4 region of 16S rRNA sequencing data	Smith et al., 2021 [22]	https://www.ncbi.nlm.nih.gov/bioproject/PRJNA714242/
V4 region of 16S rRNA sequencing data	Hurst et al., 2022 [30]	https://www.ncbi.nlm.nih.gov/bioproject/PRJNA703574/
Script		
Script used for this study	This study	https://github.com/David-microbiomics/Rscript/blob/main/Rscript_16SV4
Software and algoritms		
R version 4.2.1.	R project	https://www.r-project.org/
DADA2 R package ver. 3.17	Callahan et al., 2016	https://bioconductor.org/packages/release/bioc/html/dada2.html
Wrench R package ver. 1.20.0	Muthiah and Corrada, 2023	https://bioconductor.org/packages/release/bioc/html/Wrench.html
Phyloseq R package ver. 1.46.0	Murdie and Holmes, 2013	https://bioconductor.org/packages/release/bioc/html/phyloseq.html
Microbiome R package ver. 1.18	Lahti and Shetty,2019	https://bioconductor.org/packages/release/bioc/html/microbiome.html
Eulerr R package ver. 7.0.0	Larsson, 2022	https://www.rdocumentation.org/packages/eulerr/versions/7.0.0

Microbiomeutilities R package ver. 1.00.16	Lahti and Shetty, 2020	https://microsud.github.io/microbiomeutilities/
Ggplot2 R package ver. 3.4.4.	Wickham, 2016	https://ggplot2.tidyverse.org/
Ggstatsplot R package ver. 0.9.4	Patil, 2021	https://indrajeetpatil.github.io/ggstatsplot/
Gtsummary R package ver. 1.6.2	Sjoberg et al., 2021	https://www.danieldsjoberg.com/gtsummary/

Nasopharyngeal Swab

Nasopharyngeal swab (NPS) specimens (n = 207) were collected from adult individuals. The sample collection followed the protocol outlined by Marti et al., as detailed in N Engl J Med 2020;382:e76, DOI: 10.1056/NEJMvcm2010260. Sterile nylon-tipped swabs were used for the collection. The specimens were then placed in UTM® (Universal Transport Medium™). Duplicate samples were obtained, with one of them being subjected to SARS-CoV-2 testing at the National Institute for Respiratory Diseases in the Molecular Biology Laboratory through RT-PCR assay. The remaining nasopharyngeal sample was stored at -70°C until DNA extraction.

DNA Extraction

The samples positive for SARS-CoV-2 were centrifugated for 20 min at 14,000 RPM. The pellet was resuspended in 500 ul of Sputolysin® (Calbiochem, Germany) mixed and incubated at room temperature for 15 min. Subsequently, they were centrifuged for 5 minutes at 1500 rpm, discarding the supernatant. DNA was extracted with the QIAmp UCP pathogen miniKit (Qiagen), https://www.qiagen.com/es-us/products/discovery-and-translational-research/dna-rna-purification/dna-purification/microbial-dna/qiaamp-ucp-pathogen-mini-kit, following the mechanical pre-lysis with spin protocol for swabs, except proteinase K was extended to 20 min.

16S rRNA Amplicon PCR

Subsequently, the DNA V4 region of 16S rRNA was amplified by PCR using the primers 515f and 806r. 50 ng of template DNA was amplified with Phusion Plus Master Mix (Thermo Fisher, USA) using 2.5 µl of both forward and reverse primers [0.2µM]. PCR conditions were as follows: initial denaturation 95 °C/15 min, followed by 25 cycles of denaturation 95°C/30s, aneling 60°C/30s and polymerization 72°C/2 min. After that, final polymerization was carried out at 72 °C/10 min. PCR products were purified using DNA Clean &Concentrator (Zymo, USA).

Sequencing

Samples were pooled and barcoded following Illumina 16S metagenomics protocol (Illumina, USA), https://support.illumina.com/downloads/16s_metagenomic_sequencing_library_preparation.html. The barcodes were pooled in equimolar concentration and sequenced on the Illumina Miseq platform using the 2X250 pair-end method.

Deposited data

The raw V4 region of 16S rRNA sequencing data of this study has been deposited in the NCBI Bioproject database under accession number **PRJNA981220**.

R script

```
###############################################################################################
Modified from the DADA2 tutorial  https://benjjneb.github.io/dada2/tutorial.html
###############################################################################################
###############################################################################################
# Install DADA2
#https://benjjneb.github.io/dada2/dada-installation.html
#if (!requireNamespace("BiocManager", quietly = TRUE))
#  install.packages("BiocManager")
#BiocManager::install("dada2", version = "3.17")
#Install phyloseq
# https://joey711.github.io/phyloseq/
# Install ggplot2
# https://ggplot2.tidyverse.org/
##BiocManager::install("Wrench")
#############################################################################################

#Load packages
library(dada2)
library(phyloseq)
library(ggplot2)
library(microbiome)
library(knitr)
library(Wrench)
library(RColorBrewer)
library(ggstatsplot)
library(eulerr)


path <- "~/git_script" # CHANGE ME to the directory containing the fastq files after unzipping.
list.files(path)

#The forward and reverse read files have the following format: NAME_1.fastq.gz and
NAME_2.fastq.gz

fnFs <- sort(list.files(path, pattern="_1.fastq.gz", full.names = TRUE))
fnRs <- sort(list.files(path, pattern="_2.fastq.gz", full.names = TRUE))

# We extract the name of the sequences assuming that they have the format NAME_X.fastq.gz

sample.names <- sapply(strsplit(basename(fnFs), "_"), `[`, 1)

##QUALITY CONTROL OF SEQUENCES

plotQualityProfile(fnFs[1:6])
plotQualityProfile(fnRs[1:6])

# Assign the filenames for the filtered fastq.gz files.
filtFs <- file.path(path, "filtered", paste0(sample.names, "_F_filt.fastq.gz"))
filtRs <- file.path(path, "filtered", paste0(sample.names, "_R_filt.fastq.gz"))
names(filtFs) <- sample.names
names(filtRs) <- sample.names
```

```
# We Filter
out <- filterAndTrim(fnFs, filtFs, fnRs, filtRs, truncLen=c(242,246),
                    maxN=0, maxEE=c(2,2), truncQ=2, rm.phix=TRUE,
                    compress=TRUE, multithread=TRUE) #  Windows multithread=FALSE

head(out)

#Learn the Error Rates.

errF <- learnErrors(filtFs, multithread=TRUE)
errR <- learnErrors(filtRs, multithread=TRUE)


plotErrors(errF, nominalQ=TRUE)
plotErrors(errR, nominalQ=TRUE)

dadaFs <- dada(filtFs, err=errF, multithread = TRUE)
dadaRs <- dada(filtRs, err=errR, multithread = TRUE)

#Inspecting the returned dada-class object:
dadaFs[[1]]
dadaRs[[1]]


#Merge paired reads

mergers <- mergePairs(dadaFs, filtFs, dadaRs, filtRs, verbose=TRUE)
head(mergers[[1]])


#Construct sequence table

seqtab <- makeSequenceTable(mergers)
dim(seqtab)

table(nchar(getSequences(seqtab)))


#Remove chimeras

seqtab.nochim <- removeBimeraDenovo(seqtab, method="consensus", multithread=TRUE,
verbose=TRUE)
dim(seqtab.nochim)
sum(seqtab.nochim)/sum(seqtab)


#Track reads through the pipeline

getN <- function(x) sum(getUniques(x))
track <- cbind(out, sapply(dadaFs, getN), sapply(dadaRs, getN), sapply(mergers, getN),
rowSums(seqtab.nochim))
colnames(track) <- c("input", "filtered", "denoisedF", "denoisedR", "merged", "nonchim")
```

```
rownames(track) <- sample.names
head(track)


#Assign taxonomy

taxa <- assignTaxonomy(seqtab.nochim, "~/silva_nr99_v138.1_train_set.fa.gz", multithread=TRUE)
taxa <- addSpecies(taxa, "~/silva_species_assignment_v138.1.fa.gz")

taxa.print <- taxa
rownames(taxa.print) <- NULL
head(taxa.print)

#export files to csv to create phyloseq
setwd("~/git_script/")
trans_seqtab_nochimZ = data.frame(t(seqtab.nochim))
write.csv(trans_seqtab_nochimZ, file="asv_table.csv")
write.csv(taxa, file="tax_table.csv")


###########################################################
#We now construct a phyloseq object directly from the dada2 outputs.

###########################################################

#We assign the tables to their variables
asv_mat = asv_table
tax_mat = tax_table
samples_df = sample_data #Incorporate the metadata

#We transform to matrix
asv_mat <- as.matrix(asv_mat)
tax_mat <- as.matrix(tax_mat)

#transform to phyloseq objects
ASV = otu_table(asv_mat, taxa_are_rows = TRUE)
TAX = tax_table (tax_mat)
samples = sample_data (samples_df)

ps_omega = phyloseq(ASV, TAX, samples)

#preprocessing
#We remove the ASV with NA in Genus
ps_omega <- subset_taxa(ps_omega, !is.na(Genus))

#We consolidate by genus
(merge_omega <- tax_glom(ps_omega, taxrank="Genus"))
merge_omega

#We remove taxa that add up to less than 10 reads among all samples
ps_omega <-prune_taxa(taxa_sums(merge_omega) > 10, merge_omega)

###################### Normalization with Wrench#################
```

```
count_tab <- as.matrix(data.frame(otu_table(ps_omega)))
groupx <- sample_data(ps_omega)$COVID_severity
W <- wrench(count_tab, condition=groupx)
norm_factors <- W$nf
asv_wrench <- sweep(count_tab, 2, norm_factors, FUN = '/')
asv_wrench <-round(asv_wrench)

#the new standardized table is incorporated to phyloseq
ps_Wnorm <- ps_omega
otu_table(ps_Wnorm) <- otu_table(asv_wrench, taxa_are_rows =  TRUE)
ps_Wnorm

#ASVs that are at 0 are removed
ps_Wnorm <-prune_taxa(taxa_sums(ps_Wnorm) >0, ps_Wnorm)

##################### phyloseq with relative abundances #########

asv.relative_W <- abundances(ps_Wnorm, "compositional")
ps_W_relative <- ps_Wnorm
otu_table(ps_W_relative) <- otu_table(asv.relative_W, taxa_are_rows =  TRUE)
ps_W_relative


#################################################################

#CHARTS

#################################################################

microbiomePalette <- c('#a6cee3','#1f78b4','#b2df8a','#33a02c','#84DE02', '#FF6037',
'#aaa500','#cab2d6','#6a3d5a','#391285','#b15928',"#e41a1c","#FFA6C9", "#FF3399","#00CC99",
"#5E8C31", "#933709", "#FF7A00", "#FFDF46", "#01665e", "#35978f")

#categorization by COVID severity
samples_df$COVID_severity = factor (samples_df $ COVID_severity, levels = c ('control', "severe",
'critical'))
samples <- sample_data (samples_df)
sample_data(ps_W_relative) <- sample_data(samples)
ps_W_relative

#top 20 genus
top20 <- names(sort(taxa_sums(ps_W_relative), decreasing=TRUE))[1:20]
ps.top20 <- transform_sample_counts(ps_Wnorm, function(OTU) OTU/sum(OTU))
ps.top20 <- prune_taxa(top20, ps.top20)

p4 <- plot_bar(ps.top20, fill = "Genus") +
  facet_wrap(~COVID_severity, scales="free_x")
p4 + geom_bar(aes(color=NULL, fill= Genus), stat="identity", position="stack") +
  scale_fill_manual(values = microbiomePalette) +
  theme(panel.ontop = FALSE,
        text=element_text(size=20),
        axis.text=element_text(size=15),
        axis.title=element_text(size=15),
        plot.title=element_text(size=15),
        legend.text=element_text(size=20),
        legend.title=element_text(size=20)) +
  labs(x ="", y = "Relative abundance" )

############### Alpha diversity #################################

pseq_W = ps_Wnorm
tab_W = alpha(pseq_W, index = "all")
plot_richness(pseq_W)

############### Beta diversity ##################################

ps.prop <- transform_sample_counts(ps_Wnorm, function(otu) otu/sum(otu))
ord.nmds.bray <- ordinate(ps.prop, method="NMDS", distance="bray")
op <-plot_ordination(ps.prop, ord.nmds.bray, color="COVID_severity", title="Bray NMDS")
print(op)


###### Comparison of taxonomic groups by COVID Severity##########

#We  manually create megadata_W with, alpha diversity indices, metadata and relative abundance by
genus.

megadata_W$COVID_severity = factor (megadata_W$ COVID_severity, levels = c ('control', "severe",
'critical'))

violinpalete <- c("#4393c3", "#4d9221", "#b2182b")

q23 <- ggbetweenstats(
  data= megadata_W,
  x= "COVID_severity",
  y= "Lawsonella",
  type ="np",
  xlab= "Patients by severity",
  ylab= "Lawsonella relative abundance",
  plot.type= "boxviolin",
  centrality.plotting = FALSE,
  centrality.point.args = list (size = 5, color = "darkred"),
  centrality.label.args = list (size  = 4, nudge_x = 0.4, segment.linetype = 4,
                               min.segment.length = 0),
  ggsignif.args = list(textsize = 0, tip_length = 0.01),
  violin.args = list(scale = "width", size = 1, aes(fill = COVID_severity),alpha = 0.8),
  ggtheme = theme(panel.background = element_rect(fill = NA),
                 panel.grid.major = element_line(colour = "#ffffff"),
                 panel.ontop = FALSE,
                 text=element_text(size=15),
                 axis.text=element_text(size=15),
                 axis.title=element_text(size=15),
                 plot.title=element_text(size=10),
                 legend.text=element_text(size=100),
                 legend.title=element_text(size=10)))
q23 + scale_fill_manual(values = violinpalete)
```

```
#And the parameters were changed for each taxonomic group or index

################## Core Microbiome ################################################

disease_states <- unique(as.character(meta(ps_W_relative)$COVID_severity))
print(disease_states)

list_core <- c()

for (n in disease_states){
  ps.sub <- subset_samples(ps_W_relative, COVID_severity == n)
  core_m <- core_members(ps.sub,
                         detection = 0.001,
                         prevalence = 0.60)
  print(paste0("No. of core taxa in ", n, " : ", length(core_m)))
  list_core[[n]] <- core_m
}
print(list_core)

mycols <- c(nonCRC="#d6e2e9", CRC="#cbf3f0", H="#fcf5c7")
plot(venn(list_core),
     fills = mycols)
#################################################################################
```