#图1a没分析的有环境因子的四分位比较、去噪方法比较、筛选asv丰度、

#图1a

options(timeout = 600) # 设置超时时间为600秒

install.packages("raster")

install.packages("httr")

install.packages("geodata")

install.packages("sf")

install.packages("fmsb")

install.packages("scales")

install.packages("ggplot2")

install.packages("terra")

install.packages("png")

install.packages("grid")

install.packages("dplyr")

library(raster)

library(httr)

library(geodata)

library(sf)

library(fmsb)

library(scales)

library(ggplot2)

library(terra)

library(png)

library(grid)

library(dplyr)

# ---------------- 1. 获取指定坐标气候和植物数据 ----------------

# 获取WorldClim全球气候数据，指定保存路径

download\_path <- "D:/study/master/worldclim\_data"

climate\_data <- worldclim\_global(var = "bio", res = 2.5, path = download\_path)

climate\_data

#输入坐标

pops <- data.frame(

code = c("JR", "JJ", "TZ", "PA"),

longitude = c(119.08889, 120.19389, 121.0225, 120.35167), # 经度

latitude = c(32.13889, 32.00111, 31.99611, 28.92445) # 纬度

)

# 确保 pops 数据框中的经纬度是数值型

pops$longitude <- as.numeric(pops$longitude)

pops$latitude <- as.numeric(pops$latitude)

# 使用 sf 包转换为空间数据框

pops\_sf <- st\_as\_sf(pops, coords = c("longitude", "latitude"), crs = 4326)

pops\_sf

# 检查 pops\_sf 和WorldClim数据的 CRS

crs(pops\_sf)

crs(climate\_data)

# 查看 pops\_sf 和WorldClim的范围

ext(pops\_sf)

ext(climate\_data)

#提取坐标对应的气候数据

climate\_values <- extract(climate\_data, pops\_sf)

climate\_values

#获取植物数据

rb\_metadata <- read.delim(file = "D:/study/master/rb\_metadata.tsv", sep = "\t", header = TRUE, check.names = FALSE)

# 计算每个 Group 的数值型变量的均值

plant\_data\_mean\_id <- rb\_metadata %>%

group\_by(Origin) %>% # 按 Origin 分组

summarise(across(where(is.numeric), mean, na.rm = TRUE), .groups = "keep") %>%

mutate(Origin = factor(Origin, levels = c("JR", "TZ", "JJ", "PA"))) %>% # 自定义排序

arrange(Origin) # 按自定义顺序排列

print(plant\_data\_mean\_id)

# ---------------- 2. 绘制雷达图 ----------------

# 去除 ID 列，保留气候和植物数据

climate\_values\_data <- climate\_values[, -1]

plant\_data\_mean <- plant\_data\_mean\_id %>%

ungroup() %>% # 取消分组

select(-Origin) # 删除 Origin 列

# 最小-最大归一化函数

normalize\_data <- function(data) {

return((data - min(data)) / (max(data) - min(data)) \* 100)

}

# 对气候和植物数据进行归一化

climate\_values\_data\_normalized <- as.data.frame(lapply(climate\_values\_data, normalize\_data))

plant\_data\_mean\_normalized <- as.data.frame(lapply(plant\_data\_mean, normalize\_data))

# 定义雷达图绘制函数

create\_beautiful\_radarchart <- function(data, color = "#00AFBB",

vlabels = NULL,vlcex = 0.7,

caxislabels = NULL, title = NULL,...) {

radarchart(

data, axistype = 0, # 设置为0来完全去掉轴

# 自定义多边形

pcol = color, pfcol = scales::alpha(color, 0.5), plwd = 2, plty = 1,

# 自定义网格

cglcol = "grey", cglty = 1, cglwd = 0.8,

# 自定义轴标签

axislabcol = "grey",

# 变量标签设为NULL，去掉变量名

vlcex = 0.0007, vlabels = NULL,

caxislabels = caxislabels, title = title, ...

)

}

#气候雷达图

#保存图片

#png("D:/study/master/Main\_Figure\_tables/Figure\_1/1a\_radar\_chart\_climate.png", width = 2000, height = 2000, res = 300, bg = "transparent")

# 设置图形布局

op <- par(mar = c(1, 1, 1, 1))

par(mfrow = c(2, 2)) # 生成多个雷达图（根据采样点数量调整）

# 颜色定义

colors <- c("JR" = "#DC9445FF",

"JJ" = "#E5614CFF",

"TZ" = "#bee183",

"PA" = "#ADD8E6")

# 循环绘制每个采样点的雷达图

for(i in 1:nrow(climate\_values\_data\_normalized)) {

# 创建雷达图数据，仍然使用归一化的数据进行绘制

radar\_data <- rbind(rep(100, ncol(climate\_values\_data\_normalized)), # 最大值

rep(0, ncol(climate\_values\_data\_normalized)), # 最小值

climate\_values\_data\_normalized[i, ]) # 当前采样点的归一化数据

# 获取当前采样点的ID

sample\_id <- climate\_values$ID[i]

color <- colors[sample\_id]

# 绘制雷达图

colnames(radar\_data)<-rep("",ncol(radar\_data))# 去除变量名标签

create\_beautiful\_radarchart(

data = radar\_data,

vlabels = NULL,

caxislabels = NULL, # 去除轴标签

title = NULL, # 去掉标题

color = color

)

# 添加每个变量的标签（显示原始数据值）

for(j in 1:ncol(climate\_values\_data\_normalized)) {

# 获取每个点的位置

angle <- (2 \* pi \* (j + 3.75)) / ncol(climate\_values\_data\_normalized) #旋转角度

# 计算标签位置

x\_pos <- cos(angle) \* (as.numeric(climate\_values\_data\_normalized[i, j])+50) /130 # 设置标签距离中心的距离

y\_pos <- sin(angle) \* (as.numeric(climate\_values\_data\_normalized[i, j])+50) /130 # 设置标签距离中心的距离

# 在雷达图上添加原始数据值标签

text(x\_pos, y\_pos, labels = sprintf("%.1f", climate\_values[i, j+1]), col = "black", cex = 1)

}

}

# 关闭图形设备，保存文件

#dev.off()

#外观性状雷达图

plant\_appearance <- plant\_data\_mean[4:9]

plant\_appearance\_normalized <- plant\_data\_mean\_normalized[4:9]

#保存图片

#png("D:/study/master/Main\_Figure\_tables/Figure\_1/1a\_radar\_chart\_appearance.png", width = 2000, height = 2000, res = 300, bg = "transparent")

# 设置图形布局

op <- par(mar = c(1, 1, 1, 1))

par(mfrow = c(2, 2)) # 生成多个雷达图（根据采样点数量调整）

# 颜色定义

colors <- c("JR" = "#DC9445FF",

"JJ" = "#E5614CFF",

"TZ" = "#bee183",

"PA" = "#ADD8E6")

# 循环绘制每个采样点的雷达图

for(i in 1:nrow(plant\_appearance\_normalized)) {

# 创建雷达图数据，仍然使用归一化的数据进行绘制

radar\_data <- rbind(rep(100, ncol(plant\_appearance\_normalized)), # 最大值

rep(0, ncol(plant\_appearance\_normalized)), # 最小值

plant\_appearance\_normalized[i, ]) # 当前采样点的归一化数据

# 获取当前采样点的ID

sample\_id <- plant\_data\_mean\_id$Origin[i]

color <- colors[sample\_id]

# 绘制雷达图

colnames(radar\_data)<-rep("",ncol(radar\_data))# 去除变量名标签

create\_beautiful\_radarchart(

data = radar\_data,

vlabels = NULL,

caxislabels = NULL, # 去除轴标签

title = NULL, # 去掉标题

color = color

)

# 添加每个变量的标签（显示原始数据值）

for(j in 1:ncol(plant\_appearance\_normalized)) {

# 获取每个点的位置

angle <- (2 \* pi \* (j + 0.5)) / ncol(plant\_appearance\_normalized)#旋转角度

# 计算标签位置

x\_pos <- cos(angle) \* (as.numeric(plant\_appearance\_normalized[i, j])+50) /130 # 设置标签距离中心的距离

y\_pos <- sin(angle) \* (as.numeric(plant\_appearance\_normalized[i, j])+50) /130 # 设置标签距离中心的距离

# 在雷达图上添加原始数据值标签

text(x\_pos, y\_pos, labels = sprintf("%.1f", plant\_appearance[i, j]), col = "black", cex = 1)

}

}

# 关闭图形设备，保存文件

#dev.off()

#植物化学成分雷达图

plant\_phytochemicals <- plant\_data\_mean[10:16]

plant\_phytochemicals\_normalized <- plant\_data\_mean\_normalized[10:16]

#保存图片

#png("D:/study/master/Main\_Figure\_tables/Figure\_1/1a\_radar\_chart\_phytochemicals.png", width = 2000, height = 2000, res = 300, bg = "transparent")

# 设置图形布局

op <- par(mar = c(1, 1, 1, 1))

par(mfrow = c(2, 2)) # 生成多个雷达图（根据采样点数量调整）

# 颜色定义

colors <- c("JR" = "#DC9445FF",

"JJ" = "#E5614CFF",

"TZ" = "#bee183",

"PA" = "#ADD8E6")

# 循环绘制每个采样点的雷达图

for(i in 1:nrow(plant\_phytochemicals\_normalized)) {

# 创建雷达图数据，仍然使用归一化的数据进行绘制

radar\_data <- rbind(rep(100, ncol(plant\_phytochemicals\_normalized)), # 最大值

rep(0, ncol(plant\_phytochemicals\_normalized)), # 最小值

plant\_phytochemicals\_normalized[i, ]) # 当前采样点的归一化数据

# 获取当前采样点的ID

sample\_id <- plant\_data\_mean\_id$Origin[i]

color <- colors[sample\_id]

# 绘制雷达图

colnames(radar\_data)<-rep("",ncol(radar\_data))# 去除变量名标签

create\_beautiful\_radarchart(

data = radar\_data,

vlabels = NULL,

caxislabels = NULL, # 去除轴标签

title = NULL, # 去掉标题

color = color

)

# 添加每个变量的标签（显示原始数据值）

for(j in 1:ncol(plant\_phytochemicals\_normalized)) {

# 获取每个点的位置

angle <- (2 \* pi \* (j + 0.75)) / ncol(plant\_phytochemicals\_normalized) #旋转角度

# 计算标签位置

x\_pos <- cos(angle) \* (as.numeric(plant\_phytochemicals\_normalized[i, j])+50) /130 # 设置标签距离中心的距离

y\_pos <- sin(angle) \* (as.numeric(plant\_phytochemicals\_normalized[i, j])+50) /130 # 设置标签距离中心的距离

# 在雷达图上添加原始数据值标签

text(x\_pos, y\_pos, labels = sprintf("%.1f", plant\_phytochemicals[i, j]), col = "black", cex = 1)

}

}

# 关闭图形设备，保存文件

#dev.off()

par(mfrow = c(1, 1)) # 恢复为单个图形

#定义只有变量名标签的空白雷达图函数

create\_empty\_radarchart <- function(data, color = "#00AFBB", #vlabels = NULL,

vlcex = 0.7,

caxislabels = NULL, title = NULL,...) {

radarchart(

data, axistype = 0, # 设置为0来完全去掉轴

# 自定义多边形

pcol = color, pfcol = scales::alpha(color, 0.5), plwd = 0, plty = 1,

# 自定义网格

cglcol = "grey", cglty = 1, cglwd = 0.8,

# 自定义轴标签

axislabcol = "grey",

# 变量标签设为NULL，默认变量名

vlcex = 1.4, #vlabels = NULL,

caxislabels = caxislabels, title = title, ...

)

}

#绘制空白雷达图，忽略错误警告

#忽视错误

#try(create\_empty\_radarchart(data = climate\_values\_data\_normalized), silent = TRUE)

#png("D:/study/master/Main\_Figure\_tables/Figure\_1/1a\_empty\_radar\_chart\_appearance.png", width = 2500, height = 2500, res = 300, bg = "transparent")

#try(create\_empty\_radarchart(data = plant\_appearance\_normalized, vlabels = c("Plant height (cm)","Leaf length\n(cm)","Leaf\nwidth\n(cm)",expression("Stem diameter (cm"^2\*")"),"Bulb\nraw\nweight (g)","Bulb dry\nweight (g)")) , silent = TRUE)

#dev.off()

#png("D:/study/master/Main\_Figure\_tables/Figure\_1/1a\_empty\_radar\_chart\_phytochemicals.png", width = 2500, height = 1500, res = 300, bg = "transparent")

#try(create\_empty\_radarchart(data = plant\_phytochemicals\_normalized, vlabels = c("Soluble protein content (mg/g)","Chlorophyll a\ncontent (mg/g)","Chlorophyll\nb content\n(mg/g)","Total chlorophyll\ncontent (mg/g)","Malondialdehyde\n(nmol/g)","Peimine\ncontent\n(mg/g)","Peiminine\ncontent (mg/g)")) , silent = TRUE)

#dev.off()

# ---------------- 3. 绘制中国（江苏省和浙江省）地图 ----------------

# 下载并加载 GeoJSON 数据

geojson\_url <- "https://geo.datav.aliyun.com/areas\_v3/bound/geojson?code=100000\_full"

china\_map <- st\_read(geojson\_url)

# 可视化地图并添加坐标点（统一黑色）

chinampa\_plot <-ggplot(data = china\_map) +

geom\_sf(fill = "gray85", color = "gray85") + # 地图填充为灰色，边界为深灰色

geom\_sf(data = pops\_sf, size = 2, color = "black") + # 添加黑色点

theme\_void() + # 移除经纬度和背景网格

theme(

panel.background = element\_rect(fill = "white", color = NA), # 设置画布背景为白色

plot.background = element\_rect(fill = "white", color = NA) # 设置整个画布为白色

)

chinampa\_plot

library(ggplot2)

library(sf)

# 下载并加载江苏省（320000）和浙江省（330000）的 GeoJSON 数据

jiangsu\_geojson <- "https://geo.datav.aliyun.com/areas\_v3/bound/geojson?code=320000\_full"

zhejiang\_geojson <- "https://geo.datav.aliyun.com/areas\_v3/bound/geojson?code=330000\_full"

# 读取地图数据

jiangsu\_map <- st\_read(jiangsu\_geojson)

zhejiang\_map <- st\_read(zhejiang\_geojson)

# 合并两省地图数据

china\_jiangsu\_zhejiang\_map <- rbind(jiangsu\_map, zhejiang\_map)

# 可视化地图并添加采样点（黑色点）

china\_jiangsu\_zhejiang\_map\_plot <- ggplot() +

geom\_sf(data = china\_jiangsu\_zhejiang\_map, fill = "gray85", color = "gray50") + # 江苏浙江地图

geom\_sf(data = pops\_sf, size = 2, color = "black") + # 采样点

theme\_void() + # 移除经纬度和网格

theme(

panel.background = element\_rect(fill = "white", color = NA),

plot.background = element\_rect(fill = "white", color = NA)

)

# 显示地图

china\_jiangsu\_zhejiang\_map\_plot

#ggsave("D:/study/master/Main\_Figure\_tables/Figure\_1/1a\_china\_jiangsu\_zhejiang\_map.png", plot = china\_jiangsu\_zhejiang\_map\_plot, width = 8, height = 8, dpi = 300, bg = "transparent")

#图1b没分析的有PCoA、PERMANOVA 检验、

#图1b

if (!requireNamespace("BiocManager", quietly = TRUE))

install.packages("BiocManager")

BiocManager::install("phyloseq")

BiocManager::install("microbiome")

install.packages("devtools")

devtools::install\_github("jbisanz/qiime2R")

install.packages("vegan")

install.packages("ggplot2")

install.packages("dplyr")

install.packages("ggalluvial")

install.packages("tibble")

# 加载必要的包

library(tibble)

library(phyloseq)

library(microbiome)

library(devtools)

library(qiime2R)

library(vegan)

library(ggplot2)

library(dplyr)

library(ggalluvial)

# ---------------- 1. 合并为 phyloseq 对象 ----------------

bacteria\_ASV <- read.csv(file = "D:/study/master/meiji/bacteria\_ASV.csv",

sep=",",header=TRUE,check.names = FALSE)

# 定义文件的绝对路径（使用正斜杠）

rb\_metadata <- "D:/study/master/rb\_metadata.tsv"

# 导入元数据文件

rb\_metadata <- import\_qiime\_sample\_data(rb\_metadata)

# 查看元数据

head(rb\_metadata)

# 读取 ASV 表（特征表）

rb\_ASV <- bacteria\_ASV[, c("ASV",rb\_metadata$Sample.ID)]

rb\_ASV <- rb\_ASV %>% column\_to\_rownames(var = "ASV")

head(rb\_ASV)

# 读取 taxonomy（分类信息）

rb\_tax <- bacteria\_ASV[, 2:9]

rb\_tax <- rb\_tax %>% column\_to\_rownames(var = "ASV")

head(rb\_tax)

# 合并为 phyloseq 对象

rb\_ASV <- otu\_table(rb\_ASV, taxa\_are\_rows = TRUE)

rb\_tax <- tax\_table(as.matrix(rb\_tax))

rb\_merged <- merge\_phyloseq(rb\_ASV, rb\_tax, rb\_metadata)

# 提取 ASV 表

rb\_ASV <- otu\_table(Gmerged, taxa\_are\_rows = TRUE)

# ---------------- 3. 计算 ASV 物种丰富度 ----------------

# 计算 Observed 物种数和 Shannon 指数

alpha\_div <- estimate\_richness(rb\_merged, measures = c("Observed", "Shannon"))

# 合并元数据

alpha\_div$Sample.ID <- rownames(alpha\_div)

alpha\_div <- cbind(alpha\_div, rb\_metadata[match(rownames(alpha\_div), rb\_metadata$Sample.ID), ])

alpha\_div <- alpha\_div[, !duplicated(colnames(alpha\_div))]

alpha\_div

# 可视化 alpha 多样性

ggplot(alpha\_div, aes(x = Origin, y = Observed, fill = Origin)) +

geom\_boxplot() +

theme\_minimal() +

labs(title = "Observed Species Richness", x = "Origin", y = "Observed ASVs")

ggplot(alpha\_div, aes(x = Origin, y = Shannon, fill = Origin)) +

geom\_boxplot() +

theme\_minimal() +

labs(title = "Shannon Diversity Index", x = "Origin", y = "Shannon Index")

# ---------------- 4. 计算 ASV 丰富度的统计信息 ----------------

# 计算平均值、标准差和中位数

summary\_stats <- alpha\_div %>%

group\_by(Origin) %>%

summarise(

mean\_observed = mean(Observed),

sd\_observed = sd(Observed),

median\_observed = median(Observed),

mean\_shannon = mean(Shannon),

sd\_shannon = sd(Shannon),

median\_shannon = median(Shannon)

)

print(summary\_stats)

# ---------------- 5. NMDS 分析 (非度量多维标度) ----------------

# 计算不同分类层次的距离矩阵并进行 NMDS 分析

classification\_levels <- c("Phylum", "Class", "Order", "Family", "Genus", "Species")

# 存储每个分类层次的 NMDS 结果

nmds\_results <- list()

for (level in classification\_levels) {

# 按层次汇总数据

phylo\_agg <- tax\_glom(rb\_merged, taxrank = level)

# 计算 Bray-Curtis 距离矩阵

bray\_dist <- phyloseq::distance(phylo\_agg, method = "bray", weighted = TRUE)

# 运行 NMDS 分析

nmds <- metaMDS(bray\_dist, k = 2, trymax = 100)

# 提取 NMDS 结果并合并元数据

metadata\_df <- as.data.frame(sample\_data(phylo\_agg))

metadata\_df$Sample.ID <- rownames(metadata\_df)

nmds\_data <- as.data.frame(nmds$points)

nmds\_data$Sample.ID <- rownames(nmds\_data)

nmds\_data <- cbind(nmds\_data, metadata\_df[match(nmds\_data$Sample.ID, metadata\_df$Sample.ID), ])

nmds\_data <- nmds\_data[, !duplicated(colnames(nmds\_data))]

# 保存结果

nmds\_results[[level]] <- nmds\_data

}

# 可视化不同分类层次的 NMDS 结果

plot\_list <- list()

for (level in classification\_levels) {

nmds\_data <- nmds\_results[[level]]

p <- ggplot(nmds\_data, aes(x = MDS1, y = MDS2, color = Origin)) +

geom\_point(size = 6, alpha = 0.6, shape = 16) +

theme(

axis.text.y = element\_text(colour = "black", size = 12, face = "bold"),

axis.text.x = element\_text(colour = "black", face = "bold", size = 12),

legend.text = element\_text(size = 12, face = "bold", colour = "black"),

legend.position = "right",

legend.box.background = element\_rect(colour = "black"),

axis.title.y = element\_text(face = "bold", size = 14),

axis.title.x = element\_text(face = "bold", size = 14, colour = "black"),

legend.title = element\_text(size = 14, colour = "black", face = "bold"),

panel.background = element\_blank(),

panel.grid = element\_blank(),

axis.line = element\_line(colour = "black"),

panel.border = element\_blank(),

axis.line.x.bottom = element\_line(colour = "black", linewidth = 1.2),

axis.line.y.left = element\_line(colour = "black", linewidth = 1.2)

) +

labs(title = paste("NMDS Ordination by", level), x = "NMDS1", y = "NMDS2", colour = "Origin")

plot\_list[[level]] <- p

}

# 显示不同层次的图

plot\_list$Phylum # 可以修改为其他层次，如 "Class", "Order", etc.

# ---------------- 6. 绘制堆叠柱状图 ----------------

# 按门 (Phylum) 水平进行汇总

rb\_merged\_phylum <- tax\_glom(rb\_merged, taxrank = "Phylum")

rb\_merged\_phylum\_rel <- transform\_sample\_counts(rb\_merged\_phylum, function(x) x / sum(x))

# 获取物种相对丰度

phylum\_abundance <- psmelt(rb\_merged\_phylum\_rel)

# 可视化不同产地的物种组成

ggplot(phylum\_abundance, aes(x = Origin, y = Abundance, fill = Phylum)) +

geom\_bar(stat = "identity", position = "stack") +

theme\_minimal() +

labs(title = "Phylum-Level Composition", x = "Origin", y = "Relative Abundance") +

theme(axis.text.x = element\_text(angle = 45, hjust = 1))

# ----------------7. 绘制桑基图（Sankey Diagram） ----------------

# 计算门 (Phylum) 水平的相对丰度

rb\_merged\_phylum <- tax\_glom(rb\_merged, taxrank = "Phylum")

rb\_merged\_phylum\_rel <- transform\_sample\_counts(rb\_merged\_phylum, function(x) x / sum(x))

# 计算科 (Family) 水平的相对丰度

rb\_merged\_family <- tax\_glom(rb\_merged, taxrank = "Family")

rb\_merged\_family\_rel <- transform\_sample\_counts(rb\_merged\_family, function(x) x / sum(x))

# 提取 Phylum 级别的物种丰度

phylum\_abundance <- psmelt(rb\_merged\_phylum\_rel)

# 提取 Family 级别的物种丰度

family\_abundance <- psmelt(rb\_merged\_family\_rel)

# 选择 Family 级别的前 10 种属

top\_family <- family\_abundance %>%

group\_by(Family) %>%

summarise(mean\_abundance = mean(Abundance)) %>%

top\_n(10, mean\_abundance) %>%

pull(Family)

# 过滤数据

filtered\_family\_abundance <- family\_abundance %>%

filter(Family %in% top\_family)

# 绘制 Sankey 图

ggplot(filtered\_family\_abundance, aes(axis1 = Phylum, axis2 = Family, y = Abundance)) +

geom\_alluvium(aes(fill = Family), width = 0.5) +

geom\_stratum(width = 1/12, fill = "grey", color = "black") +

geom\_text(stat = "stratum", aes(label = after\_stat(stratum)), size = 4) +

theme\_minimal() +

labs(title = "Microbiome Composition (Phylum to Family)", x = "Taxonomic Level", y = "Relative Abundance")

#图1c

#https://zenodo.org/records/10035668/files/Tax4Fun2\_1.1.5.tar.gz?download=1

#https://zenodo.org/records/10035668/files/Tax4Fun2\_ReferenceData\_v2.tar.gz?download=1

install.packages("D:/study/master/Tax4Fun2\_1.1.5.tar.gz", repos = NULL, type = "source")

install.packages("devtools")

install.packages("remotes")

remotes::install\_github("kasperskytte/ampvis2")

library(tidyverse)

library(mgcv)

library(data.table)

library(gt)

library(vegan)

library(randomForest)

library(smacof)

library(ampvis2)

library(Maaslin2)

library(readxl)

library(ggsci)

library(ggalluvial)

library(pheatmap)

library(ggpubr)

library(geosphere)

library(here)

library(qiime2R)

library(phyloseq)

library(Tax4Fun2)

theme\_set(theme\_bw() +

theme(strip.background = element\_rect(colour = "black", fill = "white"), strip.text = element\_text(size=17, family="Helvetica")) +

theme(axis.text.x = element\_text(angle = 90, size = 18, family = "Helvetica"),

axis.text.y = element\_text(size = 18, family = "Helvetica"),

axis.title.y = element\_text(size = 18, face = "bold", family = "Helvetica"),

axis.title.x=element\_text(size = 18, face = "bold", family = "Helvetica"),

axis.ticks.x=element\_blank()) +

theme(plot.title = element\_text(hjust = 0.5, size = 20, family = "Helvetica", face = "bold")) +

theme(legend.text = element\_text(size = 18, family = "Helvetica"),

legend.title = element\_text(size = 18, family = "Helvetica", face = "bold"), legend.direction = "vertical", legend.box = "vertical"))

clean\_sample\_name <- function(raw\_name){

raw\_name = gsub('DownB', 'DN', raw\_name)

raw\_name = gsub('UpB', 'UP', raw\_name)

raw\_name = gsub('Down', 'DN', raw\_name)

clean\_name = gsub('Up', 'UP', raw\_name)

return(clean\_name)}

#读取数据

bacteria\_ASV <- read.csv(file = "D:/study/master/meiji/bacteria\_ASV.csv",

sep=",",header=TRUE,check.names = FALSE)

# 定义文件的绝对路径（使用正斜杠）

rb\_metadata <- "D:/study/master/rb\_metadata.tsv"

# 导入元数据文件

rb\_metadata <- import\_qiime\_sample\_data(rb\_metadata)

# 查看元数据

head(rb\_metadata)

# 读取 ASV 表（特征表）

rb\_ASV <- bacteria\_ASV[, c("ASV",rb\_metadata$Sample.ID)]

rb\_ASV <- rb\_ASV %>% column\_to\_rownames(var = "ASV")

head(rb\_ASV)

# 读取 taxonomy（分类信息）

rb\_tax <- bacteria\_ASV[, 2:9]

rb\_tax <- rb\_tax %>% column\_to\_rownames(var = "ASV")

head(rb\_tax)

#分步生成和读取系统发育树

tree <- read\_qza("D:/study/master/meiji/rooted-tree.qza")

phylo\_tree <- tree$data

md5<-read\_excel("D:/study/master/meiji/ASV\_md5.xlsx")#md5值

rename\_vector <- setNames(md5$`ASV ID`, md5$md5)

phylo\_tree$tip.label <- rename\_vector[phylo\_tree$tip.label] #替换tip.label

# 合并为 phyloseq 对象

rb\_ASV <- otu\_table(rb\_ASV, taxa\_are\_rows = TRUE)

rb\_tax <- tax\_table(as.matrix(rb\_tax))

rb\_merged <- merge\_phyloseq(rb\_ASV, rb\_tax, rb\_metadata, phy\_tree(phylo\_tree))

#去除非细菌

rb\_merged <- phyloseq::subset\_taxa(rb\_merged, (Kingdom!="d\_\_Eukaryota") | is.na(Kingdom))

rb\_merged <- phyloseq::subset\_taxa(rb\_merged, (Kingdom!="d\_\_Archaea") | is.na(Kingdom))

rb\_merged <- phyloseq::subset\_taxa(rb\_merged, (Order!="o\_\_Chloroplast") )

rb\_merged <- phyloseq::subset\_taxa(rb\_merged, (Family!="f\_\_Mitochondria"))

rb\_merged <- phyloseq::subset\_taxa(rb\_merged, (Family!="NA"))

# 提取 ASV 表

rb\_ASV <- as.matrix(otu\_table(rb\_merged, taxa\_are\_rows=T))

rb\_ASV\_df <- rb\_ASV[rowSums(rb\_ASV[])>0,]

rb\_ASV\_df\_table <- as.data.frame(rb\_ASV\_df)

rb\_ASV\_df\_table <- rownames\_to\_column(rb\_ASV\_df\_table, var = "ID")

write.table(rb\_ASV\_df\_table, "D:/study/master/rb\_ASV\_df\_table.txt", sep = "\t", quote = FALSE, row.names = TRUE)

#KEGG 参考基因组注释，fasta过大可以拆分再合并

runRefBlast(path\_to\_otus = 'D:/study/master/meiji/ASV\_reps.fasta',

path\_to\_reference\_data = 'D:/study/master/Tax4Fun2\_ReferenceData\_v2',

path\_to\_temp\_folder = "D:/study/master/Kelp\_Ref99NR",

database\_mode = 'Ref99NR',

use\_force = TRUE,

num\_threads = 4)

ref\_blast <- fread("D:/study/master/ref\_blast/ref\_blast\_md5.txt")

ref\_blast$V1 <- rename\_vector[ref\_blast$V1] #替换md5值

write.table(ref\_blast, "D:/study/master/Kelp\_Ref99NR/ref\_blast.txt", sep = "\t", quote = FALSE, row.names = FALSE, col.names = FALSE)

####预测群落功能####

makeFunctionalPrediction(path\_to\_otu\_table = 'D:/study/master/rb\_ASV\_df\_table.txt',

path\_to\_reference\_data = 'D:/study/master/Tax4Fun2\_ReferenceData\_v2',

path\_to\_temp\_folder = 'D:/study/master/Kelp\_Ref99NR',

database\_mode = 'Ref99NR',

normalize\_by\_copy\_number = TRUE, #默认，用参考数据库中每个序列计算的16S rRNA拷贝数的平均值进行归一化

min\_identity\_to\_reference = 0.97,

normalize\_pathways = FALSE)#默认，将把每个KO的相对丰度关联到它所属的每个路径上

kegg <- fread(file = "D:/study/master/Kelp\_Ref99NR/functional\_prediction.txt", check.names = FALSE)

library(httr)

# KEGG KO 列表的 URL

url <- "https://rest.kegg.jp/list/ko"

# 发送 GET 请求

response <- GET(url)

# 检查请求状态

if (status\_code(response) == 200) {

# 提取文本内容

content <- content(response, "text")

# 保存到文件

writeLines(content, "D:/study/master/ko\_list.txt")

} else {

print("请求失败")

}

ko\_data <- read.table("D:/study/master/ko\_list.txt", sep = "\t", header = FALSE, stringsAsFactors = FALSE)

head(ko\_data)

#图2a没分析的有筛选多于生物学重复的序列、

#图2a

# 安装 R 包

#https://github.com/grunwaldlab/metacoder/archive/refs/heads/master.zip

install.packages("remotes")

install.packages(c("ggplot2", "dplyr", "readr", "tibble", "vegan", "ape", "agricolae", "stringr"))

install.packages("BiocManager")

BiocManager::install("phyloseq")

library(remotes)

library(phyloseq)

library(metacoder)

library(ggplot2)

library(dplyr)

library(readr)

library(tibble)

library(stringr)

remotes::install\_github("mikemc/phyloseqCompanion")

remotes::install\_local("D:/study/micro/reference/metacoder-master.zip")

#细菌树状热图

bacteria\_ASV <- read.csv(file = "D:/study/master/meiji/bacteria\_ASV.csv",

sep=",",header=TRUE,check.names = FALSE)

# 定义文件的绝对路径（使用正斜杠）

metadata <- "D:/study/master/metadata.tsv"

# 导入元数据文件

metadata <- import\_qiime\_sample\_data(metadata)

# 查看元数据

head(metadata)

# 读取 ASV 表（特征表）

#b\_ASV <- bacteria\_ASV[, c("ASV",metadata$Sample.ID)]#根际细菌

b\_ASV <- bacteria\_ASV[, c(9,13:60)]#根际和鳞茎细菌

colnames(b\_ASV)

b\_ASV <- b\_ASV %>% column\_to\_rownames(var = "ASV")

head(b\_ASV)

# 读取 taxonomy（分类信息）

b\_tax <- bacteria\_ASV[, 2:9]

b\_tax <- b\_tax %>% column\_to\_rownames(var = "ASV")

head(b\_tax)

# 合并为 phyloseq 对象

b\_ASV <- otu\_table(b\_ASV, taxa\_are\_rows = TRUE)

b\_tax <- tax\_table(as.matrix(b\_tax))

b\_merged <- merge\_phyloseq(b\_ASV, b\_tax, metadata)

#去除非细菌

b\_merged <- subset\_taxa(b\_merged, (Kingdom!="d\_\_Eukaryota") | is.na(Kingdom))

b\_merged <- subset\_taxa(b\_merged, (Kingdom!="d\_\_Archaea") | is.na(Kingdom))

b\_merged <- subset\_taxa(b\_merged, (Order!="o\_\_Chloroplast") )

b\_merged <- subset\_taxa(b\_merged, (Family!="f\_\_Mitochondria"))

b\_merged <- subset\_taxa(b\_merged, (Family!="NA"))

# 提取 ASV 表

b\_ASV <- otu\_table(b\_merged, taxa\_are\_rows=TRUE)

# 提取分类表

b\_tax <- tax\_table(b\_merged)

# 提取样本元数据

metadata <- as\_tibble(sample\_data(b\_merged))

#\*\*转换 taxonomy 表为 metacoder 兼容格式\*\*

b\_tax\_df<-as.data.frame(b\_tax)

b\_tax\_df$ASV <- rownames(b\_tax\_df)

b\_tax\_df$lineage <- paste( #参考parse\_tax\_data函数参数

b\_tax\_df$Kingdom, # 添加 Kingdom

b\_tax\_df$Phylum, # 添加 Phylum

b\_tax\_df$Class, # 添加 Class

b\_tax\_df$Order, # 添加 Order

b\_tax\_df$Family, # 添加 Family

b\_tax\_df$Genus, # 添加 Genus

b\_tax\_df$Species, # 添加 Species

sep = ";" # 使用分号作为分隔符

)

b\_tax\_df$lineage <- str\_replace\_all(b\_tax\_df$lineage, ";uncultured\_bacterium", "")

b\_tax\_df\_clean <- b\_tax\_df %>%

filter(!str\_detect(Genus, "unclassified"))

b\_tax\_df\_clean <- b\_tax\_df\_clean %>%

filter(!str\_detect(Genus, "norank"))

#b\_tax\_df\_clean <- b\_tax\_df\_clean %>%

#add\_count(Genus, name = "Genus\_Count") %>% # 添加计数列，列名为Genus\_Count

#filter(Genus\_Count >= 2) %>% # 保留出现次数≥2的属

#select(-Genus\_Count) # 移除计数列（可选）

b\_ASV<-as.data.frame(b\_ASV)

b\_ASV$ASV <- rownames(b\_ASV)

b\_otus <- merge(b\_tax\_df\_clean[, c("ASV", "lineage")], b\_ASV, by = "ASV", all.x = TRUE)

row\_sums <- rowSums(b\_otus[3:48])#去除过少ASV

rows\_to\_remove <- row\_sums <= 3

b\_otus <- b\_otus[!rows\_to\_remove, ]

# \*\*解析数据，使 metacoder 识别\*\*

b\_tree <- parse\_tax\_data(b\_otus,

class\_cols = "lineage",

class\_sep = ";",

class\_regex = "^(.+)\_\_(.+)$",

class\_key = c("tax\_rank" = "taxon\_rank", "name" = "taxon\_name"))

# 移除无意义的分类名

b\_tree <- metacoder::filter\_taxa(b\_tree, taxon\_names != "")

b\_tree <- metacoder::filter\_taxa(b\_tree, taxon\_names != "Bacteria;;;;;")

#筛选属及以上

b\_tree <- b\_tree %>%

metacoder::filter\_taxa(taxon\_ranks == "g", supertaxa=T)

# 计算丰度

b\_tree$data$tax\_abund <- calc\_taxon\_abund(b\_tree, "tax\_data")

# 计算出现次数

b\_tree$data$tax\_occ <- calc\_n\_samples(b\_tree, "tax\_abund")

# \*\*绘制分类热树（Heat Tree）\*\*

set.seed(199) # 保证图形的可复现性，可循环1到1000

b\_heat\_tree <- heat\_tree(b\_tree,

#基础映射

node\_color = n\_obs, # 颜色代表 ASV 频率

node\_size = n\_obs, # 节点大小代表 ASV 频率

node\_label = NA,#taxon\_names, # 节点标签为分类名称

edge\_size= n\_obs,

#颜色配置

edge\_color\_range = c("#CC8394FF", "#AC563BFF", "#CDA97CFF","#7C8EC5FF","#2B3C51FF"),

node\_color\_range = c("#CC8394FF", "#AC563BFF", "#CDA97CFF","#7C8EC5FF","#2B3C51FF"),

edge\_color = n\_samples, # 边的颜色表示分类单元出现在多少个样本中

#图例控制

make\_node\_legend = FALSE,

make\_edge\_legend=FALSE,

edge\_legend\_title = NULL, # 标题格式

edge\_color\_axis\_label = NULL, # 隐藏左侧颜色条标签

edge\_color\_digits = 2,

#布局优化

initial\_layout = "re",

layout = "da") # 选择合适的树形布局

b\_heat\_tree

# 导出图片

# ggsave("D:/study/master/Main\_Figure\_tables/Figure\_2/2a\_bacteria\_heat\_tree.png", plot = b\_heat\_tree, width = 10, height = 10, dpi = 300)

#真菌树状热图

fungi\_ASV <- read.csv(file = "D:/study/master/meiji/fungi\_ASV.csv",

sep=",",header=TRUE,check.names = FALSE)

# 定义文件的绝对路径（使用正斜杠）

metadata <- "D:/study/master/metadata.tsv"

# 导入元数据文件

metadata <- import\_qiime\_sample\_data(metadata)

# 查看元数据

head(metadata)

# 读取 ASV 表（特征表）

#f\_ASV <- fungi\_ASV[, c("ASV",metadata$Sample.ID)]#根际真菌

f\_ASV <- fungi\_ASV[, c(9,13:60)]#根际和鳞茎真菌

colnames(f\_ASV)

f\_ASV <- f\_ASV %>% column\_to\_rownames(var = "ASV")

head(f\_ASV)

# 读取 taxonomy（分类信息）

f\_tax <- fungi\_ASV[, 2:9]

f\_tax <- f\_tax %>% column\_to\_rownames(var = "ASV")

head(f\_tax)

# 合并为 phyloseq 对象

f\_ASV <- otu\_table(f\_ASV, taxa\_are\_rows = TRUE)

f\_tax <- tax\_table(as.matrix(f\_tax))

f\_merged <- merge\_phyloseq(f\_ASV, f\_tax, metadata)

#去除非细菌

f\_merged <- subset\_taxa(f\_merged, (Kingdom!="d\_\_Eukaryota") | is.na(Kingdom))

f\_merged <- subset\_taxa(f\_merged, (Kingdom!="d\_\_Archaea") | is.na(Kingdom))

f\_merged <- subset\_taxa(f\_merged, (Order!="o\_\_Chloroplast") )

f\_merged <- subset\_taxa(f\_merged, (Family!="f\_\_Mitochondria"))

f\_merged <- subset\_taxa(f\_merged, (Family!="NA"))

# 提取 ASV 表

f\_ASV <- otu\_table(f\_merged, taxa\_are\_rows=TRUE)

# 提取分类表

f\_tax <- tax\_table(f\_merged)

# 提取样本元数据

metadata <- as\_tibble(sample\_data(f\_merged))

#\*\*转换 taxonomy 表为 metacoder 兼容格式\*\*

f\_tax\_df<-as.data.frame(f\_tax)

f\_tax\_df$ASV <- rownames(f\_tax\_df)

f\_tax\_df$lineage <- paste( #参考parse\_tax\_data函数参数

f\_tax\_df$Kingdom, # 添加 Kingdom

f\_tax\_df$Phylum, # 添加 Phylum

f\_tax\_df$Class, # 添加 Class

f\_tax\_df$Order, # 添加 Order

f\_tax\_df$Family, # 添加 Family

f\_tax\_df$Genus, # 添加 Genus

f\_tax\_df$Species, # 添加 Species

sep = ";" # 使用分号作为分隔符

)

f\_tax\_df$lineage <- str\_replace\_all(f\_tax\_df$lineage, ";uncultured\_bacterium", "")

f\_tax\_df\_clean <- f\_tax\_df %>%

filter(!str\_detect(Genus, "unclassified"))

f\_tax\_df\_clean <- f\_tax\_df\_clean %>%

filter(!str\_detect(Genus, "norank"))#clean有变化，ftree没变。

#f\_tax\_df\_clean <- f\_tax\_df\_clean %>%

#add\_count(Genus, name = "Genus\_Count") %>% # 添加计数列，列名为Genus\_Count

#filter(Genus\_Count >= 2) %>% # 保留出现次数≥2的属

#select(-Genus\_Count) # 移除计数列（可选）

f\_ASV<-as.data.frame(f\_ASV)

f\_ASV$ASV <- rownames(f\_ASV)

f\_otus <- merge(f\_tax\_df\_clean[, c("ASV", "lineage")], f\_ASV, by = "ASV", all.x = TRUE)

#row\_sums <- rowSums(f\_otus[3:48])#去除过少ASV

#rows\_to\_remove <- row\_sums <= 3

#f\_otus <- f\_otus[!rows\_to\_remove, ]

# \*\*解析数据，使 metacoder 识别\*\*

f\_tree <- parse\_tax\_data(f\_otus,

class\_cols = "lineage",

class\_sep = ";",

class\_regex = "^(.+)\_\_(.+)$",

class\_key = c("tax\_rank" = "taxon\_rank", "name" = "taxon\_name"))

# 移除无意义的分类名

f\_tree <- metacoder::filter\_taxa(f\_tree, taxon\_names != "")

f\_tree <- metacoder::filter\_taxa(f\_tree, taxon\_names != "Fungi;;;;;")

#筛选属及以上

f\_tree <- f\_tree %>%

metacoder::filter\_taxa(taxon\_ranks == "g", supertaxa=T)

# 计算丰度

f\_tree$data$tax\_abund <- calc\_taxon\_abund(f\_tree, "tax\_data")

# 计算出现次数

f\_tree$data$tax\_occ <- calc\_n\_samples(f\_tree, "tax\_abund")

# \*\*绘制分类热树（Heat Tree）\*\*

set.seed(247) # 保证图形的可复现性，可循环1到1000

f\_heat\_tree <- heat\_tree(f\_tree,

#基础映射

node\_color = n\_obs, # 颜色代表 ASV 频率

node\_size = n\_obs, # 节点大小代表 ASV 频率

node\_label = NA,#taxon\_names, # 节点标签为分类名称

edge\_size= n\_obs,

#颜色配置

edge\_color\_range = c("#CC8394FF", "#AC563BFF", "#CDA97CFF","#7C8EC5FF","#2B3C51FF"),

node\_color\_range = c("#CC8394FF", "#AC563BFF", "#CDA97CFF","#7C8EC5FF","#2B3C51FF"),

edge\_color = n\_samples, # 边的颜色表示分类单元出现在多少个样本中

#图例控制

make\_node\_legend = FALSE,

make\_edge\_legend=FALSE,

edge\_legend\_title = NULL, # 标题格式

edge\_color\_axis\_label = NULL, # 隐藏左侧颜色条标签

edge\_color\_digits = 2,

#布局优化

initial\_layout = "re",

layout = "da") # 选择合适的树形布局

f\_heat\_tree

# 导出图片

# ggsave("D:/study/master/Main\_Figure\_tables/Figure\_2/2a\_fungi\_heat\_tree.png", plot = f\_heat\_tree, width = 10, height = 10, dpi = 300)

#图2b没有分析的有分组样本asv出现次数、

#图2b

install.packages(c("ggplot2", "vegan", "knitr", "dplyr", "tibble", "iNEXT"))

if (!requireNamespace("BiocManager", quietly = TRUE))

install.packages("BiocManager")

BiocManager::install(c("phyloseq", "biomformat"))

# 加载必要的 R 包

library(phyloseq)

library(biomformat)

library(ggplot2)

library(vegan)

library(knitr)

library(dplyr)

library(iNEXT)

library(tibble)

bacteria\_ASV <- read.csv(file = "D:/study/master/meiji/bacteria\_ASV.csv",

sep=",",header=TRUE,check.names = FALSE)

# 定义文件的绝对路径（使用正斜杠）

rb\_metadata <- "D:/study/master/rb\_metadata.tsv"

# 导入元数据文件

rb\_metadata <- import\_qiime\_sample\_data(rb\_metadata)

# 查看元数据

head(rb\_metadata)

# 读取 ASV 表（特征表）

rb\_ASV <- bacteria\_ASV[, c("ASV",rb\_metadata$Sample.ID)]

rb\_ASV <- rb\_ASV %>% column\_to\_rownames(var = "ASV")

head(rb\_ASV)

# 读取 taxonomy（分类信息）

rb\_tax <- bacteria\_ASV[, 2:9]

rb\_tax <- rb\_tax %>% column\_to\_rownames(var = "ASV")

head(rb\_tax)

# 合并为 phyloseq 对象

rb\_ASV <- otu\_table(rb\_ASV, taxa\_are\_rows = TRUE)

rb\_tax <- tax\_table(as.matrix(rb\_tax))

rb\_merged <- merge\_phyloseq(rb\_ASV, rb\_tax, rb\_metadata)

#去除非细菌

rb\_merged <- subset\_taxa(rb\_merged, (Kingdom!="d\_\_Eukaryota") | is.na(Kingdom))

rb\_merged <- subset\_taxa(rb\_merged, (Kingdom!="d\_\_Archaea") | is.na(Kingdom))

rb\_merged <- subset\_taxa(rb\_merged, (Order!="o\_\_Chloroplast") )

rb\_merged <- subset\_taxa(rb\_merged, (Family!="f\_\_Mitochondria"))

rb\_merged <- subset\_taxa(rb\_merged, (Family!="NA"))

# 提取 ASV 表

rb\_ASV <- otu\_table(rb\_merged, taxa\_are\_rows=TRUE)

# 提取分类表

rb\_tax <- tax\_table(rb\_merged)

# 提取样本元数据

rb\_metadata <- as\_tibble(sample\_data(rb\_merged))

groups <- c("JRG", "JJG", "TZG", "PAG")

sample\_lists <- list()

asv\_lists <- list()

vec\_lists <- list()

sample\_data(rb\_merged)$Group <- as.factor(sample\_data(rb\_merged)$Group)

sample\_data(rb\_merged)$Sample.ID <- as.factor(sample\_data(rb\_merged)$Sample.ID)

sample\_data(rb\_merged)$Origin <- as.factor(sample\_data(rb\_merged)$Origin)

sample\_data(rb\_merged)$Niche <- as.factor(sample\_data(rb\_merged)$Niche)

for (group in groups) {

subset\_result <- subset\_samples(rb\_merged, Group == group)

# 计算每个 ASV 的出现频率

asv\_table\_df <- otu\_table(subset\_result, taxa\_are\_rows=TRUE)

asv\_lists[[group]] <- asv\_table\_df

sumrow\_group <- unname(rowSums(asv\_table\_df > 0))

sort\_group <- sort(sumrow\_group, decreasing=TRUE)

vec\_group <- sort\_group[sort\_group > 0]

vec\_lists[[group]] <- vec\_group

}

# 准备 iNEXT 输入数据

list\_exped\_all <- list(jrg=c(ncol(asv\_lists$JRG),vec\_lists$JRG),jjg=c(ncol(asv\_lists$JJG),vec\_lists$JJG), tzg=c(ncol(asv\_lists$TZG),vec\_lists$TZG), pag=c(ncol(asv\_lists$PAG),vec\_lists$PAG))

# 执行 iNEXT 分析丰度

out\_all\_exped <- iNEXT(list\_exped\_all, q=0, datatype="incidence\_freq", se=TRUE, conf=0.95, nboot=99)

# 将 iNEXT 结果转换为数据框

df <- fortify(out\_all\_exped, type = 1)

# 分割出观测数据和预测数据

df.point <- df[which(df$Method == "Observed"),]

df.line <- df[which(df$Method != "Observed"),]

df.line$Method <- factor(df.line$Method, c("Rarefaction", "Extrapolation"))

df.asympote <- data.frame(y = c(24,4),

Asymptote = c("bhg","jjg","ntg","pag"))

# 绘图

ggplot(df, aes(x=x, y=y, colour=Assemblage)) +

#geom\_point(aes(shape=Assemblage), size=5, data=df.point) +

geom\_line(aes(linetype=Method), lwd=1.5, data=df.line) +

geom\_ribbon(aes(ymin=y.lwr, ymax=y.upr, fill=Assemblage, colour=NULL), alpha=0.2) +

labs(x="Number of sequences", y="Species diversity") +

scale\_fill\_manual(values=c("#8C57A2FF", "#3EBCB6", "#97A1A7FF", "#DC9445FF")) +

scale\_color\_manual(values=c("#8C57A2FF", "#3EBCB6", "#97A1A7FF", "#DC9445FF")) +

scale\_linetype\_discrete(name="Method") +

theme\_bw() + theme(panel.border = element\_blank(), panel.grid.major = element\_blank(), panel.grid.minor = element\_blank(), axis.line = element\_line(colour = "black"))

# 计算频率比

inext\_freq\_results <- out\_all\_exped$AsyEst

inext\_freq\_results$prop <- inext\_freq\_results$Observed / inext\_freq\_results$Estimator

inext\_freq\_results <- inext\_freq\_results[inext\_freq\_results$Diversity == 'Species richness',]

# 计算中位数和四分位数

median\_GD\_freq <- inext\_freq\_results %>%

summarise(med = median(prop),

lower\_quartile = quantile(prop, 0.25),

median = quantile(prop, 0.5),

upper\_quartile = quantile(prop, 0.75))

#图2c

install.packages(

"microViz",

repos = c(davidbarnett = "https://david-barnett.r-universe.dev", getOption("repos"))

)

install.packages(c("dplyr", "tidyverse", "data.table", "hillR",

"viridis", "hrbrthemes", "paletteer", "mgcv"))

if (!requireNamespace("BiocManager", quietly = TRUE))

install.packages("BiocManager")

BiocManager::install(c("phyloseq", "breakaway", "microbiome"))

install.packages("remotes")

remotes::install\_github("kstagaman/phyloseqCompanion")

library(dplyr)

library(tidyverse)

library(breakaway)

library(data.table)

library(phyloseq)

library(microViz)

library(viridis)

library(hrbrthemes)

library(paletteer)

library(microbiome)

library(mgcv)

library(hillR)

library(phyloseqCompanion)

bacteria\_ASV <- read.csv(file = "D:/study/master/meiji/bacteria\_ASV.csv",

sep=",",header=TRUE,check.names = FALSE)

# 定义文件的绝对路径（使用正斜杠）

rb\_metadata <- "D:/study/master/rb\_metadata.tsv"

# 导入元数据文件

rb\_metadata <- import\_qiime\_sample\_data(rb\_metadata)

# 查看元数据

head(rb\_metadata)

# 读取 ASV 表（特征表）

rb\_ASV <- bacteria\_ASV[, c("ASV",rb\_metadata$Sample.ID)]

rb\_ASV <- rb\_ASV %>% column\_to\_rownames(var = "ASV")

head(rb\_ASV)

# 读取 taxonomy（分类信息）

rb\_tax <- bacteria\_ASV[, 2:9]

rb\_tax <- rb\_tax %>% column\_to\_rownames(var = "ASV")

head(rb\_tax)

# 合并为 phyloseq 对象

rb\_ASV <- otu\_table(rb\_ASV, taxa\_are\_rows = TRUE)

rb\_tax <- tax\_table(as.matrix(rb\_tax))

rb\_merged <- merge\_phyloseq(rb\_ASV, rb\_tax, rb\_metadata)

#去除非细菌

rb\_merged <- subset\_taxa(rb\_merged, (Kingdom!="d\_\_Eukaryota") | is.na(Kingdom))

rb\_merged <- subset\_taxa(rb\_merged, (Kingdom!="d\_\_Archaea") | is.na(Kingdom))

rb\_merged <- subset\_taxa(rb\_merged, (Order!="o\_\_Chloroplast") )

rb\_merged <- subset\_taxa(rb\_merged, (Family!="f\_\_Mitochondria"))

rb\_merged <- subset\_taxa(rb\_merged, (Family!="NA"))

# 提取 ASV 表

rb\_ASV\_df <- as.matrix(otu\_table(rb\_merged, taxa\_are\_rows=T))

rb\_ASV\_df <- rb\_ASV\_df[rowSums(rb\_ASV\_df[])>0,]

# 提取分类表

rb\_tax <- tax\_table(rb\_merged)

# 提取样本元数据

rb\_metadata <- as\_tibble(sample\_data(rb\_merged))

diversity\_nomis <- phyloseq::sample\_data(estimate\_richness(rb\_merged,measures=c("Observed","Shannon")))

alphadiv\_nomis <- merge\_phyloseq(rb\_merged, diversity\_nomis)

alphadt\_nomis\_df<- sample\_data(alphadiv\_nomis)

#计算丰富度

OR <- alphadt\_nomis\_df[, c("Sample.ID", "Group", "Observed")]

ASVrichness\_GFS <- OR %>%

group\_by(Group) %>%

summarise(average=mean(Observed), std=sd(Observed))

ASVrichness\_GFS

average\_richness <- OR %>%

summarise(average=mean(Observed), std=sd(Observed))

average\_richness

median\_richness <- OR %>%

summarise(median=median(Observed), x = quantile(Observed, c(0.25, 0.5, 0.75)))

median\_richness

#画图

habitat\_labeller <- c("JRG" = "Jiangsu\nJurong", "JJG" = "Jiangsu\nJingjiang","TZG" = "Jiangsu\nTongzhou","PAG" = "Zhejiang\nPanan")

plot\_OR <- ggplot(OR,aes(x= Group,y=Observed, color= Group)) +

geom\_violin(width=1.4,alpha=0.5) +

geom\_boxplot(width=0.1, color="black", alpha=1, outlier.shape=NA) +

geom\_jitter(position=position\_jitter(0.2), alpha=0.9) + # Set alpha value for transparency

theme\_bw()+

theme(axis.text.x = element\_text(angle = 90, vjust = 0.5, hjust = 1),

legend.position = "none") +

labs(y = "ASV Richness", x = "") +

scale\_x\_discrete(labels = habitat\_labeller)+

scale\_colour\_manual(values=c("#8C57A2FF", "#3EBCB6", "#97A1A7FF", "#DC9445FF"))

plot\_OR

# ggsave("images/20240226\_ASV\_richness\_violins.pdf", plot\_OR, height = 5, width = 8)

# saveRDS(plot\_OR, file = "images/20240226\_ASV\_richness\_violins.rds")

#Shannon多样性

Shannon <- alphadt\_nomis\_df[, c("Sample.ID", "Group", "Observed", "Shannon")]

Shannon\_GFS <- Shannon %>%

group\_by(Group) %>%

summarise(average=mean(Shannon), std=sd(Shannon))

Shannon\_GFS

median\_shannon <- Shannon %>%

reframe(median=median(Shannon), x = quantile(Shannon, c(0.25, 0.5, 0.75)))

median\_shannon

#画图

plot\_Shannon <- ggplot(Shannon,aes(x= Group,y=Shannon, color= Group)) +

geom\_violin(width=1.4,alpha=0.5) +

geom\_boxplot(width=0.1, color="black", alpha=1, outlier.shape=NA) +

geom\_jitter(position=position\_jitter(0.2), alpha=0.9) + # Set alpha value fShannon transparency

theme\_bw() +

theme(axis.text.x = element\_text(angle = 90,vjust = 0.5, hjust = 1),

legend.position = "none") +

scale\_x\_discrete(labels = habitat\_labeller)+

labs(x = "")+

scale\_colour\_manual(values=c("#8C57A2FF", "#3EBCB6", "#97A1A7FF", "#DC9445FF"))

plot\_Shannon

# ggsave("images/20240226\_Shannon.pdf", plot\_Shannon, height = 5, width = 8)

# saveRDS(plot\_Shannon, file = "images/20240226\_Shannon\_violins.rds")

#Hill number - q=1

hill\_q1\_nomis<-as.data.frame(hill\_taxa(t(rb\_ASV\_df), q = 1))

hill\_q1\_indices\_nomis <- phyloseq::sample\_data(hill\_q1\_nomis)

rb\_merged\_hill\_nomis <- phyloseq::merge\_phyloseq(rb\_merged,hill\_q1\_indices\_nomis)

meta\_diversity\_nomis\_hill <- sample.data.frame(rb\_merged\_hill\_nomis)

median\_shannon\_hill<- meta\_diversity\_nomis\_hill %>%

summarise(median=median(hill\_taxa.t.rb\_ASV\_df...q...1.), x = quantile(hill\_taxa.t.rb\_ASV\_df...q...1..1, c(0.25, 0.5, 0.75)))

#均匀度

bulla\_estimate <- phyloseq::sample\_data(microbiome::evenness(rb\_merged, index="all"))

alphadiv\_nomis <- merge\_phyloseq(alphadiv\_nomis, bulla\_estimate)

alphadt\_nomis\_df<- sample\_data(alphadiv\_nomis)

hist(alphadt\_nomis\_df$bulla)

evenness\_GFS <- alphadt\_nomis\_df %>%

group\_by(Group) %>%

summarise(average=mean(bulla), std=sd(bulla))

evenness\_GFS

evenness\_GFS\_median <- alphadt\_nomis\_df %>%

reframe(median=median(bulla), x = quantile(bulla, c(0.25, 0.5, 0.75)))

evenness\_GFS\_median

evenness\_GFS\_total <- alphadt\_nomis\_df %>%

summarise(average=mean(bulla), std=sd(bulla))

#提取元数据平均值

rb\_metadata\_mean <- rb\_metadata %>%

group\_by(Group) %>%

summarise(across(where(is.numeric), \(x) mean(x, na.rm = TRUE))) %>%

drop\_na()

alphadt\_nomis\_df<- sample\_data(alphadiv\_nomis)%>%

data.frame()%>%

left\_join(rb\_metadata\_mean, by = "Group")

#纬度对海拔的影响

lat\_ele <- ggplot(data=alphadt\_nomis\_df,aes(x= latitude.x,y= elevation.x))+

geom\_point()+

geom\_smooth(method="gam", formula = y ~ s(x, bs = 'tp', k = 3))+

theme\_bw() +

theme(panel.border = element\_blank(), panel.grid.major = element\_blank(),

panel.grid.minor = element\_blank(), axis.line = element\_line(colour = "black"))

lat\_ele

mod\_lat <- gam(data=alphadt\_nomis\_df, formula = elevation.x ~ s(latitude.x, bs='tp', k = 3))

summary(mod\_lat)

alphadt\_nomis\_df$ele\_resids = mod\_lat$residuals

pred <- predict(lm(data=alphadt\_nomis\_df, formula = Observed ~ ele\_resids),

se.fit = TRUE, interval = "confidence")

limits <- as.data.frame(pred$fit)

spe\_ele <- ggplot(alphadt\_nomis\_df, aes(x=ele\_resids, y=Observed)) +

geom\_point(size=3, alpha=0.4,color="#3F459BFF") +

geom\_smooth(method='lm', se=T, formula = y ~ x, fill="blue", color="black", alpha=0.2, span=0.3) +

geom\_line(aes(x = ele\_resids, y = limits$lwr),linetype = 2) +

geom\_line(aes(x = ele\_resids, y = limits$upr),linetype = 2) +

theme\_bw() + theme(panel.border = element\_blank(), panel.grid.major = element\_blank(),

panel.grid.minor = element\_blank(), axis.line = element\_line(colour = "black"))+

labs(x = "Elevation model residuals", y = "ASV richness")

spe\_ele

#ggsave("images/20240226\_richness\_elevation\_model\_residuals.pdf", spe\_ele, height = 5, width = 7)

#saveRDS(spe\_ele, file = "images/20240226\_richness\_elevation\_model\_residuals.rds")

model\_elevation<-lm(data=alphadt\_nomis\_df, formula = Observed ~ ele\_resids)

summary(model\_elevation)

#贝母素甲和海拔

pei\_ele <- ggplot(alphadt\_nomis\_df,aes(x=peimine.x,y= elevation.x))+

geom\_point()+

geom\_smooth(method="gam", formula = y ~ s(x, bs = 'tp'))+

theme\_bw() + theme(panel.border = element\_blank(), panel.grid.major = element\_blank(),

panel.grid.minor = element\_blank(), axis.line = element\_line(colour = "black"))

pei\_ele

#贝母素甲和多样性

pred\_pei <- predict(lm(alphadt\_nomis\_df, formula = Observed ~ peimine.x),

se.fit = TRUE, interval = "confidence")

limits\_pei<- as.data.frame(pred\_pei$fit)

spe\_pei <- ggplot(alphadt\_nomis\_df, aes(x=peimine.x, y=Observed)) +

geom\_point(size=3, alpha=0.4,color="#3F459BFF") +

geom\_smooth(method='lm', se=T, formula = y ~ x, fill="blue", color="black", alpha=0.2, span=0.3) +

geom\_line(aes(x = peimine.x, y = limits\_pei$lwr),linetype = 2) +

geom\_line(aes(x = peimine.x, y = limits\_pei$upr),linetype = 2) +

theme\_bw() + theme(panel.border = element\_blank(), panel.grid.major = element\_blank(),

panel.grid.minor = element\_blank(), axis.line = element\_line(colour = "black"))+

labs(x = " Peimine ()", y = "ASV richness")

spe\_pei

#ggsave("images/20240226\_richness\_peierage.pdf", spe\_pei, height = 5, width = 7)

#saveRDS(spe\_pei, file = "images/20240226\_richness\_peierage.rds")

model\_pei<-lm(data=alphadt\_nomis\_df, formula = Observed ~ peimine.x)

summary(model\_pei)

#图2d

install.packages(c("phyloseq", "vegan", "ggplot2", "ecodist"))

install.packages("tibble")

install.packages('devtools')

if (!requireNamespace("BiocManager", quietly = TRUE))

install.packages("BiocManager")

BiocManager::install("phyloseq")

install.packages("remotes")

remotes::install\_github("jbisanz/phyloseqCompanion")

library(phyloseq)

library(phyloseqCompanion)

library(vegan)

library(ggplot2)

library(ecodist)

library(tibble)

library(devtools)

install\_github("pmartinezarbizu/pairwiseAdonis/pairwiseAdonis")

bacteria\_ASV <- read.csv(file = "D:/study/master/meiji/bacteria\_ASV.csv",

sep=",",header=TRUE,check.names = FALSE)

# 定义文件的绝对路径（使用正斜杠）

rb\_metadata <- "D:/study/master/rb\_metadata.tsv"

# 导入元数据文件

rb\_metadata <- import\_qiime\_sample\_data(rb\_metadata)

# 查看元数据

head(rb\_metadata)

# 读取 ASV 表（特征表）

rb\_ASV <- bacteria\_ASV[, c("ASV",rb\_metadata$Sample.ID)]

rb\_ASV <- rb\_ASV %>% column\_to\_rownames(var = "ASV")

head(rb\_ASV)

# 读取 taxonomy（分类信息）

rb\_tax <- bacteria\_ASV[, 2:9]

rb\_tax <- rb\_tax %>% column\_to\_rownames(var = "ASV")

head(rb\_tax)

# 合并为 phyloseq 对象

rb\_ASV <- otu\_table(rb\_ASV, taxa\_are\_rows = TRUE)

rb\_tax <- tax\_table(as.matrix(rb\_tax))

rb\_merged <- merge\_phyloseq(rb\_ASV, rb\_tax, rb\_metadata)

#去除非细菌

rb\_merged <- subset\_taxa(rb\_merged, (Kingdom!="d\_\_Eukaryota") | is.na(Kingdom))

rb\_merged <- subset\_taxa(rb\_merged, (Kingdom!="d\_\_Archaea") | is.na(Kingdom))

rb\_merged <- subset\_taxa(rb\_merged, (Order!="o\_\_Chloroplast") )

rb\_merged <- subset\_taxa(rb\_merged, (Family!="f\_\_Mitochondria"))

rb\_merged <- subset\_taxa(rb\_merged, (Family!="NA"))

# 提取 ASV 表

rb\_ASV <- as.matrix(otu\_table(rb\_merged, taxa\_are\_rows=T))

rb\_ASV\_df <- rb\_ASV[rowSums(rb\_ASV[])>0,]

# 提取分类表

rb\_tax <- tax\_table(rb\_merged)

# 提取样本元数据

rb\_metadata <- as\_tibble(sample\_data(rb\_merged))

vegan\_otu <- function(physeq){

OTU <- otu\_table(physeq)

if(taxa\_are\_rows(OTU)){

OTU <- t(OTU)

}

return(as(OTU, "matrix"))

}

# 标准化单列数据（例如 latitude 列）

rb\_metadata\_nmds <- rb\_metadata

rb\_metadata\_nmds$ele\_sp <- scale(rb\_metadata\_nmds$elevation)

rb\_metadata\_nmds$ele\_attribute <- "A"

rb\_metadata\_nmds$ele\_attribute <- ifelse((rb\_metadata\_nmds$ele\_sp) < 1, rb\_metadata\_nmds$ele\_attribute == "B", rb\_metadata\_nmds$ele\_attribute == "A")

rb\_metadata\_nmds$ele\_attribute <- as.factor(rb\_metadata\_nmds$ele\_attribute)

## 画图

asv\_table\_nmds <- as.matrix((otu\_table(rb\_merged, taxa\_are\_rows=T)))

asv\_table\_nmds\_f <- asv\_table\_nmds[rowSums(asv\_table\_nmds[])>0,]

nmds\_bc\_nomis <- metaMDS(t(log1p(asv\_table\_nmds\_f)), distance = "bray", k = 2, trymax=999)

stressplot(nmds\_bc\_nomis)

nmds\_bc\_nomis$stress

data.scores = as.data.frame(scores(nmds\_bc\_nomis)$sites)

data.scores$Sample = rb\_metadata\_nmds$Sample.ID

data.scores$Group = rb\_metadata\_nmds$Group

head(data.scores)

nmds\_bc\_plot = ggplot(data.scores, aes(x = NMDS1, y = NMDS2)) +

geom\_point(size = 6, aes(colour = Group))+

#stat\_ellipse(aes(x=NMDS1, y=NMDS2,color=Group),type = "norm")+

theme(axis.text.y = element\_text(colour = "black", size = 12, face = "bold"),

axis.text.x = element\_text(colour = "black", face = "bold", size = 12),

legend.text = element\_text(size = 12, face ="bold", colour ="black"),

legend.position = "right", axis.title.y = element\_text(face = "bold", size = 14),

axis.title.x = element\_text(face = "bold", size = 14, colour = "black"),

legend.title = element\_text(size = 14, colour = "black", face = "bold"),

panel.background = element\_blank(), panel.border = element\_rect(colour = "black", fill = NA, size = 1.2),

legend.key=element\_blank()) +

labs(x = "NMDS1", colour = "Region", y = "NMDS2", shape = "Type")

nmds\_bc\_plot

colors<-c("#97A1A7FF","#bee183","#DC9445FF","#EDD03E")

plot(x=data.scores$NMDS1, y=data.scores$NMDS2, type="n", xlim = c(-2.5, 2.9), ylim = c(-1.6, 1.8))

points(nmds\_bc\_nomis, display = "sites", cex = 2.3, pch=19, col=alpha(colors[factor(data.scores$Group)], 0.8))

ordisurf(nmds\_bc\_nomis, rb\_metadata\_nmds$ele\_sp, add = TRUE, col="blue", labcex=1)

vegan\_matrix<- vegan\_otu(rb\_merged)

rb\_bray <- vegdist(log1p(vegan\_matrix), method="bray")

bdisp\_nomis<- betadisper(rb\_bray, rb\_metadata\_nmds$Group, type=c("centroid"))

bdisp\_nomis

aov\_bdisp <-anova(bdisp\_nomis)

permutest(bdisp\_nomis, pairwise=T)

bdisp\_nomis\_lat<- betadisper(rb\_bray, rb\_metadata\_nmds$latitude, type=c("centroid"))

bdisp\_nomis\_lat

aov\_bdisp\_lat <-anova(bdisp\_nomis\_lat)

permutest(bdisp\_nomis\_lat, pairwise=T)

rb\_ado <- adonis2(rb\_bray ~ Group, permutations = 999, method = "bray", data=rb\_metadata\_nmds)

rb\_ado\_latitude <- adonis2(rb\_bray ~ ele\_attribute, permutations = 999, method = "bray", data=rb\_metadata\_nmds)

library(pairwiseAdonis)

pairwise\_rb <- pairwise.adonis(rb\_bray, rb\_metadata\_nmds$Group, p.adjust.m="holm")

## Test the effects of latitude on community composition

pairwise\_rb\_lat <- pairwise.adonis(rb\_bray, rb\_metadata\_nmds$ele\_attribute, p.adjust.m="holm")

#图3之复杂核心asv，通过概率模型和热图分析，研究微生物在不同样本中出现的概率，确定“核心微生物组”

install.packages(c("reshape2", "ggplot2", "tibble", "foreach", "doParallel"))

if (!requireNamespace("BiocManager", quietly = TRUE)) {

install.packages("BiocManager")

}

BiocManager::install(c("phyloseq", "speedyseq", "phyloseqCompanion"))

library("reshape2")

library("speedyseq")

library("phyloseq")

library("phyloseqCompanion")

library(tibble)

library(foreach)

library(doParallel)

library(ggplot2)

#读取数据

bacteria\_ASV <- read.csv(file = "D:/study/master/meiji/bacteria\_ASV.csv",

sep=",",header=TRUE,check.names = FALSE)

# 定义文件的绝对路径（使用正斜杠）

rb\_metadata <- "D:/study/master/rb\_metadata.tsv"

# 导入元数据文件

rb\_metadata <- import\_qiime\_sample\_data(rb\_metadata)

# 查看元数据

head(rb\_metadata)

# 读取 ASV 表（特征表）

rb\_ASV <- bacteria\_ASV[, c("ASV",rb\_metadata$Sample.ID)]

rb\_ASV <- rb\_ASV %>% column\_to\_rownames(var = "ASV")

head(rb\_ASV)

# 读取 taxonomy（分类信息）

rb\_tax <- bacteria\_ASV[, 2:9]

rb\_tax <- rb\_tax %>% column\_to\_rownames(var = "ASV")

head(rb\_tax)

# 合并为 phyloseq 对象

rb\_ASV <- otu\_table(rb\_ASV, taxa\_are\_rows = TRUE)

rb\_tax <- tax\_table(as.matrix(rb\_tax))

rb\_merged <- merge\_phyloseq(rb\_ASV, rb\_tax, rb\_metadata)

#去除非细菌

rb\_merged <- subset\_taxa(rb\_merged, (Kingdom!="d\_\_Eukaryota") | is.na(Kingdom))

rb\_merged <- subset\_taxa(rb\_merged, (Kingdom!="d\_\_Archaea") | is.na(Kingdom))

rb\_merged <- subset\_taxa(rb\_merged, (Order!="o\_\_Chloroplast") )

rb\_merged <- subset\_taxa(rb\_merged, (Family!="f\_\_Mitochondria"))

rb\_merged <- subset\_taxa(rb\_merged, (Family!="NA"))

# 提取 ASV 表

rb\_ASV <- as.matrix(otu\_table(rb\_merged, taxa\_are\_rows=T))

rb\_ASV\_df <- rb\_ASV[rowSums(rb\_ASV[])>0,]

core\_data <- as.data.frame(sample\_data(rb\_merged)) #元数据

asv\_count\_level <- otu\_table(rb\_merged, taxa\_are\_rows=T)

asv\_count\_levelf <- asv\_count\_level[rowSums(asv\_count\_level[])>0,]#丰度样本asv

asv\_core\_ab = transform\_sample\_counts(rb\_merged, function(x) x / sum(x))#标准化丰度样本asv

asv\_core <- otu\_table(asv\_core\_ab, taxa\_are\_rows=T)

asv\_coref <- asv\_core[rowSums(asv\_core[])>0,]

tax\_core <-as.matrix(tax\_table(rb\_merged))#分类信息

metadata\_nomis <- sample.data.frame(rb\_merged)#元数据

metadata\_nomis <- sample\_data(metadata\_nomis)

# 筛选标准之预测不同阈值（多少算存在）ASV 在各地分布的概率

prob\_occ <- function(asv\_table, metadata, asv, ab\_thr){

subset = as.data.frame(t(as.matrix(asv\_table[rownames(asv\_table) == asv,])))

subset[subset < ab\_thr] = 0

subset[subset > 0] = 1

subset$Group = vapply(1:nrow(subset), function(x){as.character(metadata$Group[metadata$Sample.ID == rownames(subset)[x]])}, FUN.VALUE = character(1))

colnames(subset)[colnames(subset) == asv] = 'asv'

fit = glm(asv ~ Group, data=subset, family = binomial())

# fit = gam(asv ~ s(latitude, longitude, type='sos'), data=subset, family=binomial())

# pred\_df = metadata[,c('latitude','longitude')]

pred\_df = expand.grid(Group=unique(metadata$Group))

pred\_df$pred = predict(fit, newdata=pred\_df, type='response')

pred\_out = data.frame(ASV = asv, Ab\_thr = ab\_thr)

pred\_out$average\_pred = mean(pred\_df$pred)

for (Group in unique(metadata$Group)){

pred\_out[Group] = pred\_df$pred[pred\_df$Group == Group]}

return(pred\_out)}

seq\_log <- function(v1, v2, n) {exp(seq(from = log(v1), to = log(v2), length.out = n))}

#cl <- makeCluster(4) # 4 个核心

#registerDoParallel(cl) # 注册并行计算

#proba\_df = foreach(abundance=seq\_log(1/5000,1,30), .combine='rbind') %:%

# foreach(asv=rownames(asv\_coref), .combine='rbind') %dopar% {

# prob\_occ(asv\_coref, core\_data, asv, abundance)

# }

#stopCluster(cl) # 释放计算资源

abundance\_values <- seq\_log(1/5000, 1, 30)

proba\_df\_list <- list()

for (ab in abundance\_values) {

asv\_results <- foreach(asv = rownames(asv\_coref), .combine = 'rbind') %do% {

prob\_occ(asv\_coref, core\_data, asv, ab)

}

proba\_df\_list[[as.character(ab)]] <- asv\_results

}

proba\_df <- do.call(rbind, proba\_df\_list)

print(proba\_df) # 输出最终的结果

write.csv(proba\_df, "D:/study/master/proba\_df.csv")

proba\_df <- read.csv(file = "D:/study/master/proba\_df.csv",sep=",",header=TRUE,check.names = FALSE)

ggplot(data=proba\_df, aes(x=(Ab\_thr), y=(average\_pred), group=ASV))+

geom\_line() + scale\_x\_log10()

core\_size <- expand.grid(Abundance = seq\_log(1/5000, 1, 30),

Prevalence = seq\_log(1/5000, 1, 30))

core\_size$N = vapply(1:nrow(core\_size), function(x){length(unique(proba\_df$ASV[(proba\_df$Ab\_thr > core\_size$Abundance[x]) & (proba\_df$average\_pred > core\_size$Prevalence[x])]))}, FUN.VALUE = numeric(1))

ggplot(core\_size, aes(x=Abundance,y=Prevalence)) + geom\_tile(aes(fill=N)) +

geom\_line(data=data.frame(x=c(0,1),y=c(0,1)), aes(x=x,y=y), color='darkgrey', linetype = "dashed") + geom\_point(aes(x=0.0008685272, y=0.2), color='red',size=5) +

scale\_x\_log10() + scale\_y\_log10() + scale\_fill\_gradient(name = 'Core size', trans = "log", breaks = c(1,10,100,1000),na.value = 'transparent') + theme\_linedraw()

#ggsave('core\_microbiome\_biogeo.pdf', width=6, height=5)

format(proba\_df$Ab\_thr, scientific=F)

unique(proba\_df$Ab\_thr)

proba\_df\_thre <- filter(proba\_df[,-1], near(proba\_df$Ab\_thr, 0.0011650220))

binary\_transform <- sapply(proba\_df\_thre[,4:7], function(x) ifelse(x > 0.03846154, TRUE, FALSE),

USE.NAMES = F)

binary\_transform\_merge <- merge(x=binary\_transform, y=proba\_df\_thre, by="row.names")

binary\_sum <- binary\_transform\_merge %>%

rowwise()%>% mutate(sum=sum(c\_across(JJG.x:PAG.x)))%>%filter(sum>2)#多于一半

binary\_tax\_merge <- merge(x=binary\_sum, y=tax\_core, by.x="ASV", by.y=0)

merge\_core\_abundance <- merge(asv\_coref, binary\_tax\_merge, by.x="row.names", by.y="ASV" )

merge\_core\_abundance <- merge\_core\_abundance %>% select(-c(Row.names.y, JJG.x, TZG.x, JRG.x, PAG.x, Ab\_thr, average\_pred, JJG.y, TZG.y, JRG.y, PAG.y, sum, Kingdom, Phylum, Class, Order, Family, Genus, Species))

unAsIs <- function(X) {#去除AsIs类

if("AsIs" %in% class(X)) {

class(X) <- class(X)[-match("AsIs", class(X))]

}

X

}

merge\_core\_abundance$Row.names<-unAsIs(merge\_core\_abundance$Row.names)

rownames(merge\_core\_abundance) <- merge\_core\_abundance$Row.names

merge\_core\_abundance$Row.names <- NULL

merge\_core\_abundance\_m <- as.matrix(merge\_core\_abundance)

merge\_core\_abundance\_final <- otu\_table(merge\_core\_abundance\_m, taxa\_are\_rows=T)

tax\_NOMIS <- tax\_table(tax\_core)

merged\_NOMIS\_core\_ab<- merge\_phyloseq(merge\_core\_abundance\_final, tax\_NOMIS, metadata\_nomis)

merged\_NOMIS\_core\_ab\_otu <- (as.matrix(otu\_table(merged\_NOMIS\_core\_ab, taxa\_are\_rows=T)))

melt\_asv <- melt(merged\_NOMIS\_core\_ab\_otu)

merge\_asv\_data <- merge(as.data.frame(melt\_asv),as.matrix(metadata\_nomis), by.x="Var2",by.y="Sample.ID")

sum\_mr <- merge\_asv\_data %>% group\_by(Group)%>% summarize(summrr=sum(value), n=n\_distinct(Var2))%>% summarize(ar\_mr=summrr/n, Group)

median(sum\_mr$ar\_mr)

quantile(sum\_mr$ar\_mr, prob=c(.25,.5,.75), type=2)

#图3a

install.packages(c("RColorBrewer", "ggplot2", "tidyverse", "reshape2", "ggpubr"))

install.packages("phyloseqCompanion")

if (!requireNamespace("BiocManager", quietly = TRUE)) {

install.packages("BiocManager")

}

BiocManager::install(c("phyloseq", "indicspecies"))

library("RColorBrewer")

library("ggplot2")

library(tidyverse)

library(reshape2)

library(ggpubr)

library(phyloseq)

library(indicspecies)

library(phyloseqCompanion)

#读取数据

bacteria\_ASV <- read.csv(file = "D:/study/master/meiji/bacteria\_ASV.csv",

sep=",",header=TRUE,check.names = FALSE)

# 定义文件的绝对路径（使用正斜杠）

rb\_metadata <- "D:/study/master/rb\_metadata.tsv"

# 导入元数据文件

rb\_metadata <- import\_qiime\_sample\_data(rb\_metadata)

# 查看元数据

head(rb\_metadata)

# 读取 ASV 表（特征表）

rb\_ASV <- bacteria\_ASV[, c("ASV",rb\_metadata$Sample.ID)]

rb\_ASV <- rb\_ASV %>% column\_to\_rownames(var = "ASV")

head(rb\_ASV)

# 读取 taxonomy（分类信息）

rb\_tax <- bacteria\_ASV[, 2:9]

rb\_tax <- rb\_tax %>% column\_to\_rownames(var = "ASV")

head(rb\_tax)

# 合并为 phyloseq 对象

rb\_ASV <- otu\_table(rb\_ASV, taxa\_are\_rows = TRUE)

rb\_tax <- tax\_table(as.matrix(rb\_tax))

rb\_merged <- merge\_phyloseq(rb\_ASV, rb\_tax, rb\_metadata)

#去除非细菌

rb\_merged <- subset\_taxa(rb\_merged, (Kingdom!="d\_\_Eukaryota") | is.na(Kingdom))

rb\_merged <- subset\_taxa(rb\_merged, (Kingdom!="d\_\_Archaea") | is.na(Kingdom))

rb\_merged <- subset\_taxa(rb\_merged, (Order!="o\_\_Chloroplast") )

rb\_merged <- subset\_taxa(rb\_merged, (Family!="f\_\_Mitochondria"))

rb\_merged <- subset\_taxa(rb\_merged, (Family!="NA"))

# 提取 ASV 表

rb\_ASV <- as.matrix(otu\_table(rb\_merged, taxa\_are\_rows=T))

rb\_ASV\_df <- rb\_ASV[rowSums(rb\_ASV[])>0,]

#核心asv

# 计算 ASV 相对丰度

rb\_ASV\_rel <- sweep(rb\_ASV\_df, 2, colSums(rb\_ASV\_df), "/")

# 设定相对丰度阈值 0.1% (即 0.001)

asv\_above\_threshold <- rb\_ASV\_rel >= 0.001

# 将样本元数据与列名匹配

sample\_groups <- rb\_metadata$Group

names(sample\_groups) <- rb\_metadata$.sample

# 确定每个 ASV 出现在哪些 Group 中

asv\_groups <- apply(asv\_above\_threshold, 1, function(x) unique(sample\_groups[x]))

# 计算 ASV 出现的 Group 数

asv\_group\_counts <- sapply(asv\_groups, length)

# 筛选出在至少 3 个 Group 中出现的 ASVs

core\_asvs\_names <- names(asv\_group\_counts[asv\_group\_counts >= 3])

# 提取核心 ASV 的相对丰度数据

core\_asv\_data <- rb\_ASV\_rel[core\_asvs\_names, ]

# 输出核心 ASV 数量

cat("核心 ASV 数量:", length(core\_asvs\_names), "\n")

#生成核心ASV的phyloseq

merge\_core\_abondance <- rb\_ASV\_df[rownames(rb\_ASV\_df) %in% rownames(core\_asv\_data),]

mca\_table <- otu\_table(merge\_core\_abondance, taxa\_are\_rows=T)

merged\_NOMIS\_core\_ab<- merge\_phyloseq(mca\_table, rb\_tax, rb\_metadata)

#最特殊asv

metadata\_nomis <- sample.data.frame(rb\_merged)

asv\_df <- as.data.frame(t(otu\_table(rb\_ASV\_df, taxa\_are\_rows=T)))

howmanyasv<- as.data.frame(colSums(asv\_df != 0))

colnames(howmanyasv) <- c("Count\_nb")

howmanyasv$ASV <- rownames(howmanyasv)

rownames(howmanyasv) <- NULL

asvdfmelt <- melt(as.matrix(asv\_df))#长宽表格转换

asvdfmelt <- asvdfmelt[asvdfmelt$value >0,]

endemic\_one <- howmanyasv %>%

filter(Count\_nb == 1)

merge\_endemic\_abundance <- rb\_ASV\_df[rownames(rb\_ASV\_df) %in% endemic\_one$ASV,]

#比较特殊asv

control\_asv <- asvdfmelt

control\_asv$Group <- vapply((control\_asv$Var1), function(x) metadata\_nomis$Group[metadata\_nomis$Sample.ID == x], FUN.VALUE = character(1))

controlasv\_endemic <- control\_asv %>% group\_by(Group,Var2) %>% summarize(prev=n()) %>%

ungroup()%>% group\_by(Var2) %>% mutate(n=n())%>% filter(n==1)

merge\_controlasv\_abundance <- rb\_ASV\_df[rownames(rb\_ASV\_df) %in% controlasv\_endemic$Var2,]

# 指示性asv，运行 multipatt 进行指示性分析

indicator\_multipatt <- multipatt(t(rb\_ASV\_df), rb\_metadata$Group,

func = "r.g", control = how(nperm = 999))

summary(indicator\_multipatt)

# 获取显著的指示 ASV（p < 0.05）

indicator\_names <- indicator\_multipatt$sign[which(indicator\_multipatt$sign$p.value < 0.05), ]

print(indicator\_names)

#生成指示性ASV的phyloseq

merge\_indicator\_abundance <- rb\_ASV\_df[rownames(rb\_ASV\_df) %in% rownames(indicator\_names),]

indicator\_table <- otu\_table(merge\_indicator\_abundance, taxa\_are\_rows=T)

#统一读取

dat <- as.data.frame(otu\_table(rb\_merged, taxa\_are\_rows=T))

cores <- as.data.frame(otu\_table(merged\_NOMIS\_core\_ab))

endemic <- as.data.frame(otu\_table(merge\_controlasv\_abundance, taxa\_are\_rows=T))#比较特殊

indicator <- as.data.frame(indicator\_table)

uniq <- as.data.frame(otu\_table(merge\_endemic\_abundance, taxa\_are\_rows=T))#最特殊

ASV <- rownames(dat)

ASV <- as.data.frame(ASV)

cores$ASV<-rownames(cores)

rownames(cores)<-NULL

cores <- cores %>%

select(ASV)%>%

mutate(type = "core")

endemic$ASV<-rownames(endemic)

rownames(endemic)<-NULL

endemic <- endemic %>%

select(ASV)%>%

mutate(type = "endemic")

colnames(endemic) <- c("ASV", "type")

indicator$ASV<-rownames(indicator)

rownames(indicator)<-NULL

indicator <- indicator %>%

select(ASV)%>%

mutate(type = "indicator")

colnames(indicator) <- c("ASV","type")

uniq$ASV<-rownames(uniq)

rownames(uniq)<-NULL

uniq <- uniq %>%

select(ASV)%>%

mutate(type = "uniq")

colnames(uniq) <- c("ASV","type")

dat\_numb <- rbind(cores, endemic, uniq, indicator)

ASV\_other <- ASV %>%

filter(!(ASV %in% dat\_numb$ASV))%>%

mutate(type = "Other")

datToPlot <- dat\_numb %>%

arrange(ASV, type) %>%#排序

group\_by(ASV)%>%

summarise(type = str\_c(type, collapse="\_"))%>%#合并相同asv的不同标签

ungroup()%>%

bind\_rows(ASV\_other)%>%#合并asv

group\_by(type)%>%

summarise(sum = n())

datToPlot$type <- factor(datToPlot$type, levels = c("core","core\_indicator", "indicator","endemic\_indicator", "endemic\_uniq", "endemic", "Other"))

colors <- c("#034e7b", "#a6bddb", "#addd8e",

"#feb24c", "#fc4e2a", "#b10026", "#737373")

names(colors) <- c("core","core\_indicator", "indicator",

"endemic\_indicator", "endemic\_uniq", "endemic", "Other")

datToPlot <- datToPlot[order(datToPlot$type),]#排序

datToPlot$fraction = datToPlot$sum / sum(datToPlot$sum)#计算百分比

datToPlot$ymax = cumsum(datToPlot$fraction)#计算累积百分比

datToPlot$ymin = c(0, head(datToPlot$ymax, n=-1))#累积减少

datToPlot$labelPosition <- (datToPlot$ymax + datToPlot$ymin) / 2#计算标签的中心点

p1 <- ggplot(datToPlot, aes(ymax=ymax, ymin=ymin, xmax=4, xmin=3, fill=type)) +

geom\_rect() +

geom\_label( x=4, aes(y=labelPosition, label=type), size=6) +

scale\_fill\_manual(values = colors) +

coord\_polar(theta="y") +#矩形再转换为极坐标系的环形图（甜甜圈图）

#xlim(c(2, 4)) +

xlim(c(1, 4)) +

theme\_void() +

theme(legend.position = "none")

p1

dat\_sum <- rowSums(dat[1:24])

dat\_sum <- as.data.frame(dat\_sum)

dat\_sum$ASV <- rownames(dat\_sum)

datToPlotCov <- dat\_numb %>%

arrange(ASV, type) %>%

group\_by(ASV)%>%

summarise(type = str\_c(type, collapse="\_"))%>%

ungroup()%>%

bind\_rows(ASV\_other)%>%

left\_join(dat\_sum)%>%#合并总丰度

filter(!(dat\_sum == 0))%>%

mutate(perc = dat\_sum / sum(dat\_sum))%>%#计算相对丰度

group\_by(type)%>%

summarise(fraction = sum(perc))

sum(datToPlotCov$fraction)

datToPlotCov$type <- factor(datToPlotCov$type, levels = c("core","core\_indicator", "indicator","endemic\_indicator", "endemic\_uniq", "endemic", "Other"))

datToPlotCov <- datToPlotCov[order(datToPlotCov$type),]

# datToPlotCov$fraction = datToPlotCov$sum / sum(datToPlotCov$sum)

datToPlotCov$ymax = cumsum(datToPlotCov$fraction)

datToPlotCov$ymin = c(0, head(datToPlotCov$ymax, n=-1))

datToPlotCov$labelPosition <- (datToPlotCov$ymax + datToPlotCov$ymin) / 2

p2 <- ggplot(datToPlotCov, aes(ymax=ymax, ymin=ymin, xmax=4, xmin=3, fill=type)) +

geom\_rect() +

geom\_label(x=4, aes(y=labelPosition, label=type), size=6) +

scale\_fill\_manual(values = colors) +

coord\_polar(theta="y") +

#xlim(c(2, 4)) +

xlim(c(1, 4)) +

theme\_void() +

theme(legend.position = "none")

p2

p3 <- ggarrange(p1, p2)

#图3b

install.packages("devtools")

install.packages(c("tidyverse", "RColorBrewer"))

if (!requireNamespace("BiocManager", quietly = TRUE)) {

install.packages("BiocManager")

}

BiocManager::install(c("phyloseq"))

devtools::install\_github("benjjneb/speedyseq")

devtools::install\_github("joey711/phyloseqCompanion")

library(speedyseq)

library(phyloseq)

library(tidyverse)

library(phyloseqCompanion)

library(RColorBrewer)

#读取数据

bacteria\_ASV <- read.csv(file = "D:/study/master/meiji/bacteria\_ASV.csv",

sep=",",header=TRUE,check.names = FALSE)

# 定义文件的绝对路径（使用正斜杠）

rb\_metadata <- "D:/study/master/rb\_metadata.tsv"

# 导入元数据文件

rb\_metadata <- import\_qiime\_sample\_data(rb\_metadata)

# 查看元数据

head(rb\_metadata)

# 读取 ASV 表（特征表）

rb\_ASV <- bacteria\_ASV[, c("ASV",rb\_metadata$Sample.ID)]

rb\_ASV <- rb\_ASV %>% column\_to\_rownames(var = "ASV")

head(rb\_ASV)

# 读取 taxonomy（分类信息）

rb\_tax <- bacteria\_ASV[, 2:9]

rb\_tax <- rb\_tax %>% column\_to\_rownames(var = "ASV")

head(rb\_tax)

# 合并为 phyloseq 对象

rb\_ASV <- otu\_table(rb\_ASV, taxa\_are\_rows = TRUE)

rb\_tax <- tax\_table(as.matrix(rb\_tax))

rb\_merged <- merge\_phyloseq(rb\_ASV, rb\_tax, rb\_metadata)

#去除非细菌

rb\_merged <- subset\_taxa(rb\_merged, (Kingdom!="d\_\_Eukaryota") | is.na(Kingdom))

rb\_merged <- subset\_taxa(rb\_merged, (Kingdom!="d\_\_Archaea") | is.na(Kingdom))

rb\_merged <- subset\_taxa(rb\_merged, (Order!="o\_\_Chloroplast") )

rb\_merged <- subset\_taxa(rb\_merged, (Family!="f\_\_Mitochondria"))

rb\_merged <- subset\_taxa(rb\_merged, (Family!="NA"))

# 提取 ASV 表

rb\_ASV <- as.matrix(otu\_table(rb\_merged, taxa\_are\_rows=T))

rb\_ASV\_df <- rb\_ASV[rowSums(rb\_ASV[])>0,]

row\_sums <- rowSums(rb\_ASV\_df)

metadata\_nomis <- sample.data.frame(rb\_merged)

#每个asv出现在多少样本

asv\_df <- as.data.frame(t(otu\_table(rb\_ASV\_df, taxa\_are\_rows=T)))

#计算每个 ASV 在所有样本中出现的次数

howmanyasv<- as.data.frame(colSums(asv\_df != 0))

colnames(howmanyasv) <- c("Count\_nb")

howmanyasv$ASV <- rownames(howmanyasv)

rownames(howmanyasv) <- NULL

asvdfmelt <- melt(as.matrix(asv\_df))#长宽表格转换

asvdfmelt <- asvdfmelt[asvdfmelt$value >0,]

#最特殊asv

endemic\_one <- howmanyasv %>%

filter(Count\_nb == 1)

endemic\_one\_n <- howmanyasv %>% # 62661

filter(Count\_nb == 1) %>%

summarise(n = n())

endemic\_one\_n/length(unique(row.names(rb\_ASV\_df)))

merge\_equalone <- merge(endemic\_one, asvdfmelt, by.x="ASV",by.y="Var2")#合并

merge\_equalone$Group <- vapply((merge\_equalone$Var1), function(x) metadata\_nomis$Group[metadata\_nomis$Sample.ID == x], FUN.VALUE = character(1))#加列

colnames(merge\_equalone) <- c("ASV","prev","SampleID","nb\_count","Group")#asv名字、出现在所有样本中的样本次数、样本名、asv出现在某一样本的asv次数、组名

endemic\_Group\_prev <- merge\_equalone %>% group\_by(Group) %>% summarize(prev=sum(prev))

endemic\_one\_plot <- merge\_equalone %>% group\_by(Group)

endemic\_one\_plot<- endemic\_one\_plot[c("Group","ASV","prev")]

endemic\_one\_plot$Color <- "C\_uniquetoone"

prop\_unique\_Group<- endemic\_Group\_prev%>%

group\_by(Group)%>%

summarize(prop=prev/endemic\_one\_n)

colnames(prop\_unique\_Group) <- c("group\_range","prop\_unique")

#出现2-3次且比较特殊的asv

twoandnine<- howmanyasv %>% filter(Count\_nb > 1 & Count\_nb < 4)

twoandnine\_n <- twoandnine %>% summarize(n())

twoandnine\_n/length(unique(row.names(rb\_ASV\_df)))

merge\_twoandnine <- merge(twoandnine, asvdfmelt, by.x="ASV",by.y="Var2")

merge\_twoandnine$Group <- vapply((merge\_twoandnine$Var1), function(x) metadata\_nomis$Group[metadata\_nomis$Sample.ID == x], FUN.VALUE = character(1))

colnames(merge\_twoandnine) <- c("ASV","prev","SampleID","nb\_count","Group")

#计算每个asv在每个组的样本中出现的次数prev

twoandnine\_end<- merge\_twoandnine %>% group\_by(Group,ASV) %>% summarize(prev=n())

twoandnine\_end$Color <- "B\_twoandnine"

#计算每个asv在分组中出现的次数

endemism\_twoandnine <- twoandnine\_end %>% group\_by(ASV) %>% mutate(n=n()) %>% filter(n==1)%>%

ungroup()%>% group\_by(Group) %>%summarize(number=n())#都是计算asv个数

endemism\_twoandnine\_plot <- twoandnine\_end %>% group\_by(ASV) %>% mutate(n=n()) %>% filter(n==1)

sum\_endemic\_twoandnine <- endemism\_twoandnine %>% summarize(sum=sum(number))

sum\_endemic\_twoandnine/length(unique(row.names(rb\_ASV\_df)))

prop\_unique\_twonine<- endemism\_twoandnine%>%

group\_by(Group)%>%

summarize(prop=number/twoandnine\_n)

colnames(prop\_unique\_twonine)<-c("group\_range", "prop\_unique")

#大于等于10

tenandmore<- howmanyasv %>% filter(Count\_nb >= 4)

tenandmore\_n <- tenandmore %>% summarize(n())

tenandmore\_n/length(unique(row.names(rb\_ASV\_df)))

merge\_tenandmore <- merge(tenandmore, asvdfmelt, by.x="ASV",by.y="Var2")

merge\_tenandmore$Group <- vapply((merge\_tenandmore$Var1), function(x) metadata\_nomis$Group[metadata\_nomis$Sample.ID == x], FUN.VALUE = character(1))

colnames(merge\_tenandmore) <- c("ASV","prev","SampleID","nb\_count","Group")

tenandmore\_end<- merge\_tenandmore %>% group\_by(Group,ASV) %>% summarize(prev=sum(nb\_count>0))

tenandmore\_end$Color <-"A\_tenandemore"

endemism\_tenandmore <- tenandmore\_end %>% group\_by(ASV) %>% mutate(n=n()) %>% filter(n==1)%>%

ungroup()%>% group\_by(Group) %>%summarize(number=n())

endemism\_tenandmore\_plot <- tenandmore\_end %>% group\_by(ASV) %>% mutate(n=n()) %>% filter(n==1)

sum(endemism\_tenandmore$number)

#所有只出现在一个分组的asv

control\_asv <- asvdfmelt

control\_asv$Group <- vapply((control\_asv$Var1), function(x) metadata\_nomis$Group[metadata\_nomis$Sample.ID == x], FUN.VALUE = character(1))

controlasv\_end<- control\_asv %>% group\_by(Group,Var2) %>% summarize(prev=n()) %>%

ungroup()%>% group\_by(Var2) %>% mutate(n=n())%>% filter(n==1)%>%

ungroup()%>% group\_by(Group) %>%summarize(number=n())

controlasv\_total<- control\_asv %>% group\_by(Group,Var2) %>% summarize(sumi=sum(n()))%>%

ungroup()%>% group\_by(Group)%>%summarize(sumii=sum(n()))

prop\_endemic\_Group <- merge(controlasv\_end, controlasv\_total, by="Group")

prop\_endemic\_Group$prop\_endemic <- prop\_endemic\_Group$number/prop\_endemic\_Group$sumii

endemic\_one\_n/sum(controlasv\_end$number)

endemic\_one\_plot$n <- 1

#合并最特殊和比较特殊asv

df\_full <- rbind(endemic\_one\_plot, endemism\_tenandmore\_plot, endemism\_twoandnine\_plot)

niveaux <- c("JRG", "JJG", "TZG", "PAG")

df\_full$Group<- factor(df\_full$Group, levels = niveaux)

df\_full$Group <- fct\_rev(df\_full$Group)

ggplot(df\_full, aes(fill=Color, y=Group)) +

geom\_bar(position="stack", stat="count")+theme\_minimal()

merge\_asv\_endemic<- merge(t(asv\_df), df\_full, by.x="row.names",by.y="ASV")

merge\_asv\_endemic<- as.data.frame(merge\_asv\_endemic[,-c(26:29)])

row.names(merge\_asv\_endemic)<-merge\_asv\_endemic$Row.names

merge\_asv\_endemic$Row.names <- NULL

endemic\_table <- as.matrix(otu\_table(merge\_asv\_endemic, taxa\_are\_rows=T))

endemic\_table <- endemic\_table[rowSums(endemic\_table[])>0,]

merge\_endemic\_phylo <- merge\_phyloseq(endemic\_table, rb\_metadata, rb\_tax)

sample\_data(merge\_endemic\_phylo)$Group <- as.factor(sample\_data(merge\_endemic\_phylo)$Group)

sample\_data(merge\_endemic\_phylo)$Sample.ID <- as.factor(sample\_data(merge\_endemic\_phylo)$Sample.ID)

sample\_data(merge\_endemic\_phylo)$Origin <- as.factor(sample\_data(merge\_endemic\_phylo)$Origin)

sample\_data(merge\_endemic\_phylo)$Niche <- as.factor(sample\_data(merge\_endemic\_phylo)$Niche)

endemic\_taxglom <- tax\_glom(merge\_endemic\_phylo, taxrank=rank\_names(merge\_endemic\_phylo)[5], NArm=F)# 聚合结果仍是 phyloseq 对象，但 ASV 变成 Family 级别

transf\_endemic <- transform\_sample\_counts(endemic\_taxglom, function(x) x / sum(x))#相对丰度

sample\_merge\_region <- merge\_samples(transf\_endemic, "Group")#按组合并

region\_endemic <- transform\_sample\_counts(sample\_merge\_region, function(x) x / sum(x))

TopASV\_f <- names(sort(taxa\_sums(region\_endemic), TRUE)[1:19])# 选择前 19 个丰度最高的 ASV名称

top15\_NOMIS\_f <- prune\_taxa(TopASV\_f, region\_endemic)#只保留前19

top15\_NOMIS\_f <- prune\_taxa(taxa\_sums(top15\_NOMIS\_f)>0, top15\_NOMIS\_f)

top\_family<-as.data.frame(tax\_table(top15\_NOMIS\_f))

endemic\_df <- psmelt(region\_endemic)

endemic\_df$Family<- as.character(endemic\_df$Family)

endemic\_df$Family[!(endemic\_df$Family %in% top\_family$Family)] <- "Other"

endemic\_df$Family[(endemic\_df$Family == "")] <- "Other"

endemic\_df$Family[grepl("f\_\_unclassified|^$", endemic\_df$Family)] <- "Other"

endemic\_df$Family[(endemic\_df$Family == "g\_\_uncultured")] <- "Other"

n <- 20

qual\_col\_pals = brewer.pal.info[brewer.pal.info$category == 'qual',]#颜色

col\_vector = unlist(mapply(brewer.pal, qual\_col\_pals$maxcolors, rownames(qual\_col\_pals)))

#前20组成差异柱状图

barplot\_biogeo <- ggplot(data=endemic\_df, aes(x=Sample, y=Abundance, fill=Family))

barplot\_biogeo + geom\_bar(aes(), stat="identity", position="stack") +

scale\_fill\_manual(values = c("#7FC97F", "#BEAED4", "#FDC086", "#FFFF99", "#386CB0", "#F0027F", "#BF5B17", "#666666", "#1B9E77", "#D95F02", "#7570B3", "#E7298A","#66A61E",

"#E6AB02","#A6761D", "#666666", "#A6CEE3", "#1F78B4", "#B2DF8A", "#33A02C", "#FB9A99", "#E31A1C", "#FDBF6F", "#FF7F00", "#CAB2D6", "#6A3D9A",

"#FFFF99")) +

theme(legend.position="bottom") + guides(fill=guide\_legend(nrow=5))

rb\_abondance= transform\_sample\_counts(rb\_merged, function(x) x / sum(x))

rb\_asv <- otu\_table(rb\_abondance, taxa\_are\_rows=T)

rb\_asv <- rb\_asv[rowSums(rb\_asv[])>0,]

merge\_endemic\_abondance <- merge(df\_full, rb\_asv, by.x="ASV", by.y="row.names")

merge\_endemic\_abondance<- as.data.frame(merge\_endemic\_abondance[,-c(2:5)])

row.names(merge\_endemic\_abondance)<-merge\_endemic\_abondance$ASV

merge\_endemic\_abondance$ASV <- NULL

endemic\_table\_abondance <- otu\_table(merge\_endemic\_abondance, taxa\_are\_rows=T)

merge\_endemic\_abondance\_phylo <- merge\_phyloseq(endemic\_table\_abondance, rb\_tax, rb\_metadata)

end\_ab\_phylo\_table <- (as.matrix(otu\_table(merge\_endemic\_abondance\_phylo, taxa\_are\_rows=T)))

melt\_asv <- melt(end\_ab\_phylo\_table)

merge\_asv\_data <- merge(as.data.frame(melt\_asv), as.matrix(metadata\_nomis), by.x="Var2",by.y="Sample.ID")

sum\_mr <- merge\_asv\_data %>% group\_by(Group)%>% summarize(summrr=sum(value), n=n\_distinct(Var2))%>% summarize(ar\_mr=summrr/n, Group)

mean(sum\_mr$ar\_mr)

sd(sum\_mr$ar\_mr)

median\_endemism\_ab <- sum\_mr %>%

summarise(median=median(ar\_mr), x = quantile(ar\_mr, c(0.25, 0.5, 0.75)))

median\_endemism\_ab

filtered\_merge\_asv = merge\_asv\_data[merge\_asv\_data$value > 0,]

abasv\_spe <- filtered\_merge\_asv %>% group\_by(Group)%>%summarize(med\_asv=median(value), n=n\_distinct(Var2))

endemic\_taxglom\_phyla <- tax\_glom(merge\_endemic\_phylo, taxrank=rank\_names(merge\_endemic\_phylo)[2], NArm=F)#门

tax\_table\_end\_phyla <- tax\_table(endemic\_taxglom\_phyla)

transf\_endemic\_phyla = transform\_sample\_counts(endemic\_taxglom\_phyla, function(x) x / sum(x))

endemic\_region = merge\_samples(transf\_endemic\_phyla, "Group")

trans\_ra\_end\_phy = transform\_sample\_counts(endemic\_region, function(x) x / sum(x))

asv\_endemic\_phy <- otu\_table(trans\_ra\_end\_phy, taxa\_are\_rows=T)

melt\_endemic\_phy <- psmelt(asv\_endemic\_phy)

merge\_taxo\_endemic <- merge(melt\_endemic\_phy, tax\_table\_end\_phyla, by.x="OTU", by.y="row.names")

sumtot\_endo\_phylum <- merge\_taxo\_endemic %>% group\_by(Phylum) %>% summarize(sum=sum(Abundance/10))%>%

filter(!(Phylum %in% c(""," g\_\_uncultured"))) %>%

slice\_max(n=15, order\_by=sum) # 取丰度总和最高的前 15 个 'Phylum'

endemic\_taxglom\_genera <- tax\_glom(merge\_endemic\_phylo, taxrank=rank\_names(merge\_endemic\_phylo)[6], NArm=F)#属

tax\_table\_end\_genera <- tax\_table(endemic\_taxglom\_genera)

transf\_endemic\_genera = transform\_sample\_counts(endemic\_taxglom\_genera, function(x) x / sum(x))

endemic\_region\_genera = merge\_samples(transf\_endemic\_genera, "Group")

trans\_ra\_end\_gen = transform\_sample\_counts(endemic\_region\_genera, function(x) x / sum(x))

asv\_endemic\_gen <- otu\_table(trans\_ra\_end\_gen, taxa\_are\_rows=T)

melt\_endemic\_gen <- psmelt(asv\_endemic\_gen)

merge\_taxo\_endemic\_gen <- merge(melt\_endemic\_gen, tax\_table\_end\_genera, by.x="OTU", by.y="row.names")

sumtot\_endo\_genera <- merge\_taxo\_endemic\_gen %>% group\_by(Genus) %>% summarize(sum=sum(Abundance/10))%>%

filter(!(Genus %in% c(""," g\_\_uncultured"))) %>%

slice\_max(n=15, order\_by=sum)

phylumfac = factor(tax\_table(rb\_merged)[, "Phylum"])

classfac = factor(tax\_table(rb\_merged)[, "Class"])

orderfac = factor(tax\_table(rb\_merged)[, "Order"])

familyfac = factor(tax\_table(rb\_merged)[, "Family"])

genusfac = factor(tax\_table(rb\_merged)[, "Genus"])

#图3c

install.packages(c("tidyverse", "ggrepel", "RColorBrewer", "readr"))

install.packages("BiocManager")

BiocManager::install("phyloseq")

install.packages("remotes")

remotes::install\_github("kevinwolz/hisafer")

library(tidyverse)

library(ggrepel)

library(RColorBrewer)

library(phyloseq)

library(readr)

library(hisafer)

#读取数据

bacteria\_ASV <- read.csv(file = "D:/study/master/meiji/bacteria\_ASV.csv",

sep=",",header=TRUE,check.names = FALSE)

# 定义文件的绝对路径（使用正斜杠）

rb\_metadata <- read\_tsv("D:/study/master/rb\_metadata.tsv")

colnames(rb\_metadata)[1]<-"SampleID"

# 查看元数据

head(rb\_metadata)

# 读取 ASV 表（特征表）

rb\_ASV <- bacteria\_ASV[, c("ASV",rb\_metadata$SampleID)]

rb\_ASV <- rb\_ASV %>% column\_to\_rownames(var = "ASV")

head(rb\_ASV)

# 读取 taxonomy（分类信息）

rb\_tax <- bacteria\_ASV[, 2:9]

rb\_tax <- rb\_tax %>% column\_to\_rownames(var = "ASV")

head(rb\_tax)

# 提取 ASV 表

rb\_ASV\_df <- rb\_ASV[rowSums(rb\_ASV[])>0,]

rb\_ASV\_df$asv<-rownames(rb\_ASV\_df)

rb\_tax$asv<-rownames(rb\_tax)

#按属分组

dat\_m <- rb\_ASV\_df %>%

left\_join(rb\_tax)%>%

group\_by(Genus)%>%

filter(!(Genus == "g\_\_uncultured"))%>%

summarise\_if(is.numeric, sum)%>%

na.omit()%>%

pivot\_longer(cols = !Genus)%>%

left\_join(rb\_metadata, by = c("name" = "SampleID"))

sum\_all <- sum(dat\_m$value)

num\_all <- length(unique(dat\_m$name))

#计算属相对丰度（菌/总菌）

dat\_abun <- dat\_m %>%

group\_by(Genus)%>%

summarise(Abundance = sum(value)/sum\_all)

colnames(dat\_abun) <- c("Genus", "Abundance")

#计算每个属的普遍性（样本/总样本）

dat\_prev <- dat\_m %>%

group\_by(Genus, name)%>%

filter(value > 0)%>%

ungroup()%>%

group\_by(Genus)%>%

summarise(Prevalence = n()/num\_all)

tax\_sel <- rb\_tax%>%

select(-c(asv,Species))%>%

distinct()%>%

filter(Genus %in% dat\_abun$Genus)

#结合属、相对丰度、普遍性、分类信息

dat\_final <- dat\_abun%>%

left\_join(dat\_prev)%>%

left\_join(tax\_sel)

classToPlot <- dat\_final %>%

group\_by(Class)%>%

summarise(sum= sum(Abundance))%>%

arrange(desc(sum))%>%

top\_n(9)

dat\_final %>%

dplyr::arrange(desc(Abundance)) %>%

.[1:8, ] -> text\_size

dat\_final <- dat\_final%>%

mutate(Class = if\_else(Class %in% classToPlot$Class, Class, "Other"))

colors <- c(brewer.pal(9, "Set1"), "black")

names(colors) <- c(classToPlot$Class, "Other")

#相对丰度和普遍性散点图

p1 <- ggplot(dat\_final, aes(x = Prevalence, y = log10(Abundance), color = Class))+

geom\_jitter(width = 0.05, height = 0.05) + # 增加水平方向和垂直方向的抖动幅度

geom\_point()+

scale\_y\_continuous(limits = c(-7,0))+#, breaks = c(1e-5, 1e-4, 1e-3, 1e-2,1e-1, 1))+

scale\_color\_manual(values = colors)+

geom\_text\_repel(data = text\_size, aes(label = Genus), size=3)+

theme\_minimal()

#p1

#ggsave\_fitmax("PrevalenceAbundanceNOMIS\_Genus.pdf", maxwidth = 10, p1)

dat\_sum <- dat\_final %>%

select(Abundance, Prevalence)%>%

group\_by(Prevalence)%>%

reframe(sum = sum(Abundance))

#线图

p2 <- ggplot(dat\_sum, aes(x = Prevalence, y = log10(sum)))+

geom\_line()+

scale\_y\_continuous(limits = c(-7,0))+

theme\_minimal()+

geom\_smooth(method = "gam")

p2

breaks <- (0:10)/10

dat\_cut <- dat\_final%>%

mutate( ints = cut(Prevalence ,breaks = 40)) %>%

group\_by(ints) %>%

summarise(sum = sum(Abundance))

dat\_cut$Prevalence <- as.numeric(dat\_cut$ints)

p3 <- ggplot(dat\_cut, aes(x = as.numeric(ints), y = log10(sum)))+

geom\_line()+

scale\_y\_continuous(limits = c(-7,0))+

theme\_bw()

p3

temp <- dat\_final %>%

group\_by(Prevalence)%>%

summarise(sum = sum(Abundance))

temp$cumsum <- cumsum(temp$sum)#累计和

p4 <- ggplot(dat\_final,aes(x = Prevalence, y = log10(Abundance)))+

geom\_jitter(aes(color = Class), width = 0.05, height = 0.05) + # 增加水平方向和垂直方向的抖动幅度

geom\_point(aes(color = Class))+

scale\_y\_continuous(limits = c(-7,0))+#, breaks = c(1e-5, 1e-4, 1e-3, 1e-2,1e-1, 1))+

scale\_color\_manual(values = colors)+

geom\_text\_repel(data = text\_size, aes(label = Genus), size=3)+

geom\_line(data = temp, aes(x = Prevalence, y = log10(cumsum)))+

# geom\_smooth(data = dat\_sum, aes(x = Prevalence, y = log10(sum)), method = "gam", se = F)+

theme\_minimal()

#p4

#ggsave\_fitmax("D:/PrevalenceAbundanceNOMISLineGenus.pdf",maxwidth = 10, p4)

#图3d

install.packages("remotes")

remotes::install\_github("mikemc/speedyseq")

if (!requireNamespace("BiocManager", quietly = TRUE))

install.packages("BiocManager")

BiocManager::install(c("phyloseq", "phyloseqCompanion"))

install.packages(c("tidyverse", "ggridges"))

install.packages("adiv")

install.packages("vegan")

install.packages("indicspecies")

install.packages("RColorBrewer")

if(!require(devtools)) install.packages("devtools")

devtools::install\_github("kassambara/ggpubr")

library(ggpubr)

library(RColorBrewer)

library(indicspecies)

library(speedyseq)

library(phyloseq)

library(phyloseqCompanion)

library(tidyverse)

library(ggridges)

library(adiv)

library(vegan)

#读取数据

bacteria\_ASV <- read.csv(file = "D:/study/master/meiji/bacteria\_ASV.csv",

sep=",",header=TRUE,check.names = FALSE)

# 定义文件的绝对路径（使用正斜杠）

rb\_metadata <- "D:/study/master/rb\_metadata.tsv"

# 导入元数据文件

rb\_metadata <- import\_qiime\_sample\_data(rb\_metadata)

# 查看元数据

head(rb\_metadata)

# 读取 ASV 表（特征表）

rb\_ASV <- bacteria\_ASV[, c("ASV",rb\_metadata$Sample.ID)]

rb\_ASV <- rb\_ASV %>% column\_to\_rownames(var = "ASV")

head(rb\_ASV)

# 读取 taxonomy（分类信息）

rb\_tax <- bacteria\_ASV[, 2:9]

rb\_tax <- rb\_tax %>% column\_to\_rownames(var = "ASV")

head(rb\_tax)

# 合并为 phyloseq 对象

rb\_ASV <- otu\_table(rb\_ASV, taxa\_are\_rows = TRUE)

rb\_tax <- tax\_table(as.matrix(rb\_tax))

rb\_merged <- merge\_phyloseq(rb\_ASV, rb\_tax, rb\_metadata)

#去除非细菌

rb\_merged <- subset\_taxa(rb\_merged, (Kingdom!="d\_\_Eukaryota") | is.na(Kingdom))

rb\_merged <- subset\_taxa(rb\_merged, (Kingdom!="d\_\_Archaea") | is.na(Kingdom))

rb\_merged <- subset\_taxa(rb\_merged, (Order!="o\_\_Chloroplast") )

rb\_merged <- subset\_taxa(rb\_merged, (Family!="f\_\_Mitochondria"))

rb\_merged <- subset\_taxa(rb\_merged, (Family!="NA"))

# 提取 ASV 表

rb\_ASV <- as.matrix(otu\_table(rb\_merged, taxa\_are\_rows=T))

rb\_ASV\_df <- rb\_ASV[rowSums(rb\_ASV[])>0,]

#比较特殊/specific（只在某一组中发现，但可存在于该组的多个样本）、最特殊/unique（只在所有组所有样本的单个样本中发现）、核心/core（在4 组中至少有 3组的至少 1 个样本中检测到，相对丰度 ≥ 0.1%）、指示性

#核心

# 计算 ASV 相对丰度

rb\_ASV\_rel <- sweep(rb\_ASV\_df, 2, colSums(rb\_ASV\_df), "/")

# 设定相对丰度阈值 0.1% (即 0.001)

asv\_above\_threshold <- rb\_ASV\_rel >= 0.001

# 将样本元数据与列名匹配

sample\_groups <- rb\_metadata$Group

names(sample\_groups) <- rb\_metadata$.sample

# 确定每个 ASV 出现在哪些 Group 中

asv\_groups <- apply(asv\_above\_threshold, 1, function(x) unique(sample\_groups[x]))

# 计算 ASV 出现的 Group 数

asv\_group\_counts <- sapply(asv\_groups, length)

# 筛选出在至少 3 个 Group 中出现的 ASVs

core\_asvs\_names <- names(asv\_group\_counts[asv\_group\_counts >= 3])

# 提取核心 ASV 的相对丰度数据

core\_asv\_data <- rb\_ASV\_rel[core\_asvs\_names, ]

# 输出核心 ASV 数量

cat("核心 ASV 数量:", length(core\_asvs\_names), "\n")

# 可视化核心 ASV（可选）

if(requireNamespace("pheatmap", quietly = TRUE)) {

pheatmap::pheatmap(core\_asv\_data, main = "Core ASVs Heatmap")

} else {

message("pheatmap 包未安装，跳过可视化")

}

#生成核心ASV的phyloseq

merge\_core\_abondance <- rb\_ASV\_df[rownames(rb\_ASV\_df) %in% rownames(core\_asv\_data),]

mca\_table <- otu\_table(merge\_core\_abondance, taxa\_are\_rows=T)

merged\_NOMIS\_core\_ab<- merge\_phyloseq(mca\_table, rb\_tax, rb\_metadata)

#比较特殊

# 获取 ASV 在哪些 Group 中出现

asv\_presence <- apply(rb\_ASV\_df > 0, 1, function(x) unique(rb\_metadata$Group[x]))

# 筛选仅出现在单个 Group 的 ASVs

specific\_asvs\_names <- names(asv\_presence)[sapply(asv\_presence, length) == 1]

# 查看结果

specific\_asvs

#生成比较特殊ASV的phyloseq

merge\_Specific\_abundance <- rb\_ASV\_df[rownames(rb\_ASV\_df) %in% specific\_asvs\_names,]

asv\_Specific\_table <- otu\_table(merge\_Specific\_abundance, taxa\_are\_rows=T)

merge\_NOMIS\_Specific\_ab <- merge\_phyloseq(asv\_Specific\_table, rb\_tax, rb\_metadata)

#指示性

# 使用 dbMANOVAspecies 计算指示性 ASV

#indicator\_names <- dbMANOVAspecies(

# t(rb\_ASV\_df),

# rb\_metadata$Group, # 分组信息（来自元数据）

#nrep = 999, # 置换次数

#method = "BrayCurtis",# Bray-Curtis 差异矩阵

# padj = "BH" # 使用 Benjamini-Hochberg 校正

#)

# 筛选显著的指示性 ASV

# indicator\_asvs <- indicator\_results[indicator\_names$p.value < 0.05, ]

# 运行 multipatt 进行指示性分析

indicator\_multipatt <- multipatt(t(rb\_ASV\_df), rb\_metadata$Group,

func = "r.g", control = how(nperm = 999))

summary(indicator\_multipatt)

# 获取显著的指示 ASV（p < 0.05）

indicator\_names <- indicator\_multipatt$sign[which(indicator\_multipatt$sign$p.value < 0.05), ]

print(indicator\_names)

#生成指示性ASV的phyloseq

merge\_indicator\_abundance <- rb\_ASV\_df[rownames(rb\_ASV\_df) %in% rownames(indicator\_names),]

indicator\_table <- otu\_table(merge\_indicator\_abundance, taxa\_are\_rows=T)

indicator\_table\_filtered <- subset(indicator\_table, !(row.names(indicator\_table) %in% c(row.names(asv\_Specific\_table), row.names(mca\_table))))

merge\_indicator\_phylo <- merge\_phyloseq(indicator\_table\_filtered, rb\_tax, rb\_metadata)

indicator\_ASV <- row.names(indicator\_table\_filtered)

Specific\_ASV <- row.names(asv\_Specific\_table)

core\_ASV <- row.names(mca\_table)

df\_core\_Specific\_indicator <- c(indicator\_ASV,Specific\_ASV,core\_ASV)

df\_core\_Specific\_indicator\_unique <- unique(df\_core\_Specific\_indicator)

nomis\_asv\_count\_unique <- rb\_ASV\_df[rownames(rb\_ASV\_df) %in% df\_core\_Specific\_indicator\_unique,]

asv\_nomis\_unique <- otu\_table(nomis\_asv\_count\_unique, taxa\_are\_rows=T)

nomis\_asv\_unique\_phylo <- merge\_phyloseq(asv\_nomis\_unique, rb\_tax, rb\_metadata)

#属相对丰度

unique\_Genus\_taxglom <- tax\_glom(nomis\_asv\_unique\_phylo, taxrank=rank\_names(nomis\_asv\_unique\_phylo)[6], NArm=F)

transf\_unique = transform\_sample\_counts(unique\_Genus\_taxglom, function(x) x / sum(x))

#门相对丰度

unique\_phylum\_taxglom <- tax\_glom(nomis\_asv\_unique\_phylo, taxrank=rank\_names(nomis\_asv\_unique\_phylo)[2], NArm=F)

transf\_unique\_phylum = transform\_sample\_counts(unique\_phylum\_taxglom, function(x) x / sum(x))

#科相对丰度

unique\_family\_taxglom <- tax\_glom(nomis\_asv\_unique\_phylo, taxrank=rank\_names(nomis\_asv\_unique\_phylo)[5], NArm=F)

transf\_family = transform\_sample\_counts(unique\_family\_taxglom, function(x) x / sum(x))

#找出丰度最高的 11 个 Family

TopASV\_f <- names(sort(taxa\_sums(transf\_family), TRUE)[1:11])

top25\_NOMIS\_f <- prune\_taxa(TopASV\_f, transf\_unique)

top25\_NOMIS\_f <- prune\_taxa(taxa\_sums(top25\_NOMIS\_f)>0, top25\_NOMIS\_f)

top\_Genus<-as.data.frame(tax\_table(top25\_NOMIS\_f))

#属科门样本asv和分类信息

asv\_table\_all\_genus <- otu\_table(transf\_unique, taxa\_are\_rows=T)

tax\_table\_all\_genus <- tax\_table(transf\_unique)

asv\_table\_all\_family <- otu\_table(transf\_family, taxa\_are\_rows=T)

tax\_table\_all\_family <- tax\_table(transf\_family)

asv\_table\_all\_phylum <- otu\_table(transf\_unique\_phylum, taxa\_are\_rows=T)

tax\_table\_all\_phylum <- tax\_table(transf\_unique\_phylum)

#核心asv

sample\_data(merged\_NOMIS\_core\_ab)$Group <- as.factor(sample\_data(merged\_NOMIS\_core\_ab)$Group)

sample\_data(merged\_NOMIS\_core\_ab)$Sample.ID <- as.factor(sample\_data(merged\_NOMIS\_core\_ab)$Sample.ID)

sample\_data(merged\_NOMIS\_core\_ab)$Origin <- as.factor(sample\_data(merged\_NOMIS\_core\_ab)$Origin)

sample\_data(merged\_NOMIS\_core\_ab)$Niche <- as.factor(sample\_data(merged\_NOMIS\_core\_ab)$Niche)

core\_RA = transform\_sample\_counts(merged\_NOMIS\_core\_ab, function(x) x / sum(x))

core\_RA = merge\_samples(core\_RA, "Group")

core\_RA = transform\_sample\_counts(core\_RA, function(x) x / sum(x))

data\_core <- psmelt(core\_RA)

data\_core$Genus <-as.character(data\_core$Genus)

#筛选已知

sumtot\_core <- data\_core %>%

group\_by(Genus) %>%

summarize(sum = sum(Abundance)) %>%

filter(Genus %in% top\_Genus$Genus) %>%

filter(!(Genus %in% c("", "g\_\_uncultured"))) %>%

filter(!grepl("g\_\_unclassified|uncultured|unidentified|metagenome", Genus))

data\_core$Genus[!(data\_core$Genus %in% sumtot\_core$Genus)] <- "Other"

data\_core$core <- "core"

data\_core$totalAbundance <- sum(data\_core$Abundance)

data\_core\_mod <- data\_core%>%

group\_by(Genus, core)%>%

summarise(abundance = sum(Abundance)/totalAbundance)%>%

distinct()

#选择颜色

n <- 20

qual\_col\_pals = brewer.pal.info[brewer.pal.info$category == 'qual', ]

col\_vector = unlist(mapply(brewer.pal, qual\_col\_pals$maxcolors, rownames(qual\_col\_pals)))

#画图

barplot\_biogeo\_core <- ggplot(data=data\_core\_mod, aes(x=core, y=abundance, fill=Genus))

barplot\_biogeo\_core <- barplot\_biogeo\_core + geom\_bar(aes(), stat="identity", position="stack") +

scale\_fill\_manual(values = c("#4E79A7FF", "#A0CBE8FF", "#F28E2BFF", "#FFBE7DFF", "#59A14FFF", "#8CD17DFF", "#B6992DFF",

"#F1CE63FF" ,"#499894FF", "#86BCB6FF", "#E15759FF", "#FF9D9AFF", "#79706EFF", "#BAB0ACFF","#B07AA1FF", "#D4A6C8FF", "#9D7660FF", "#D7B5A6FF" )) +

theme(legend.position="bottom") + guides(fill=guide\_legend(nrow=5))

barplot\_biogeo\_core<- barplot\_biogeo\_core+ theme(panel.grid.major = element\_blank(), panel.grid.minor = element\_blank(),

panel.background = element\_blank(), axis.line = element\_line(colour = "black"))

#比较特殊asv（框架同上）

sample\_data(merge\_NOMIS\_Specific\_ab)$Group <- as.factor(sample\_data(merge\_NOMIS\_Specific\_ab)$Group)

sample\_data(merge\_NOMIS\_Specific\_ab)$Sample.ID <- as.factor(sample\_data(merge\_NOMIS\_Specific\_ab)$Sample.ID)

sample\_data(merge\_NOMIS\_Specific\_ab)$Origin <- as.factor(sample\_data(merge\_NOMIS\_Specific\_ab)$Origin)

sample\_data(merge\_NOMIS\_Specific\_ab)$Niche <- as.factor(sample\_data(merge\_NOMIS\_Specific\_ab)$Niche)

Specific\_RA = transform\_sample\_counts(merge\_NOMIS\_Specific\_ab, function(x) x / sum(x))

Specific\_RA = merge\_samples(Specific\_RA, "Group")

Specific\_RA = transform\_sample\_counts(Specific\_RA, function(x) x / sum(x))

data\_Specific <- psmelt(Specific\_RA)

data\_Specific$Genus <-as.character(data\_Specific$Genus)

sumtot\_Specific <-

data\_Specific %>% group\_by(Genus) %>% summarize(sum = sum(Abundance)) %>%

filter(Genus %in% top\_Genus$Genus) %>% filter(!(Genus %in% c(""," g\_\_uncultured"))) %>%

filter(!grepl("g\_\_unclassified|uncultured|unidentified|metagenome", Genus))

data\_Specific$Genus[!(data\_Specific$Genus %in% sumtot\_Specific$Genus)] <- "Other"

data\_Specific$Specific <- "Specific"

data\_Specific$totalAbundance <- sum(data\_Specific$Abundance)

data\_Specific\_mod <- data\_Specific%>%

group\_by(Genus, Specific)%>%

summarise(abundance = sum(Abundance)/totalAbundance)%>%

distinct()

n <- 20

qual\_col\_pals = brewer.pal.info[brewer.pal.info$category == 'qual', ]

col\_vector = unlist(mapply(brewer.pal, qual\_col\_pals$maxcolors, rownames(qual\_col\_pals)))

barplot\_biogeo\_Specific <- ggplot(data=data\_Specific\_mod, aes(x=Specific, y=abundance, fill=Genus))

barplot\_biogeo\_Specific <- barplot\_biogeo\_Specific + geom\_bar(aes(), stat="identity", position="stack") +

scale\_fill\_manual(values = c("#4E79A7FF", "#A0CBE8FF", "#F28E2BFF", "#FFBE7DFF", "#59A14FFF", "#8CD17DFF", "#B6992DFF",

"#F1CE63FF" ,"#499894FF", "#86BCB6FF", "#E15759FF", "#FF9D9AFF", "#79706EFF", "#BAB0ACFF","#B07AA1FF", "#D4A6C8FF", "#9D7660FF", "#D7B5A6FF" )) +

theme(legend.position="bottom") + guides(fill=guide\_legend(nrow=5))

barplot\_biogeo\_Specific<- barplot\_biogeo\_Specific + theme(panel.grid.major = element\_blank(), panel.grid.minor = element\_blank(),

panel.background = element\_blank(), axis.line = element\_line(colour = "black"))

#指示性asv（框架同上）

sample\_data(merge\_indicator\_phylo)$Group <- as.factor(sample\_data(merge\_indicator\_phylo)$Group)

sample\_data(merge\_indicator\_phylo)$Sample.ID <- as.factor(sample\_data(merge\_indicator\_phylo)$Sample.ID)

sample\_data(merge\_indicator\_phylo)$Origin <- as.factor(sample\_data(merge\_indicator\_phylo)$Origin)

sample\_data(merge\_indicator\_phylo)$Niche <- as.factor(sample\_data(merge\_indicator\_phylo)$Niche)

indicator\_RA = transform\_sample\_counts(merge\_indicator\_phylo, function(x) x / sum(x))

indicator\_RA = merge\_samples(indicator\_RA, "Group")

indicator\_RA = transform\_sample\_counts(indicator\_RA, function(x) x / sum(x))

data\_indicator <- psmelt(indicator\_RA)

data\_indicator$Genus <-as.character(data\_indicator$Genus)

sumtot\_indicator <-

data\_indicator %>% group\_by(Genus) %>% summarize(sum = sum(Abundance)) %>%

filter(Genus %in% top\_Genus$Genus) %>%filter(!(Genus %in% c(""," g\_\_uncultured")))%>%

filter(!grepl("g\_\_unclassified|uncultured|unidentified|metagenome", Genus))

data\_indicator$Genus[!(data\_indicator$Genus %in% sumtot\_indicator$Genus)] <- "Other"

data\_indicator$indicator <- "indicator"

data\_indicator$totalAbundance <- sum(data\_indicator$Abundance)

data\_indicator\_mod <- data\_indicator%>%

group\_by(Genus, indicator)%>%

summarise(abundance = sum(Abundance)/totalAbundance)%>%

distinct()

n <- 20

qual\_col\_pals = brewer.pal.info[brewer.pal.info$category == 'qual', ]

col\_vector = unlist(mapply(brewer.pal, qual\_col\_pals$maxcolors, rownames(qual\_col\_pals)))

barplot\_biogeo\_indicator <- ggplot(data=data\_indicator\_mod, aes(x=indicator, y=abundance, fill=Genus))

barplot\_biogeo\_indicator <- barplot\_biogeo\_indicator + geom\_bar(aes(), stat="identity", position="stack") +

scale\_fill\_manual(values = c("#4E79A7FF", "#A0CBE8FF", "#F28E2BFF", "#FFBE7DFF", "#59A14FFF", "#8CD17DFF", "#B6992DFF",

"#F1CE63FF" ,"#499894FF", "#86BCB6FF", "#E15759FF", "#FF9D9AFF", "#79706EFF", "#BAB0ACFF","#B07AA1FF", "#D4A6C8FF", "#9D7660FF", "#D7B5A6FF" )) +

theme(legend.position="bottom") + guides(fill=guide\_legend(nrow=5))

barplot\_biogeo\_indicator <- barplot\_biogeo\_indicator + theme(panel.grid.major = element\_blank(), panel.grid.minor = element\_blank(),

panel.background = element\_blank(), axis.line = element\_line(colour = "black"))

#合并3图

ggarrange(barplot\_biogeo\_Specific,barplot\_biogeo\_core,barplot\_biogeo\_indicator, ncol = 3, nrow = 1, common.legend=T)

dataset\_Specific\_filter <- as.data.frame(data\_Specific[data\_Specific$Abundance >0,])

rename\_Specific <- rename(dataset\_Specific\_filter, category = Specific)

dataset\_core\_filter <- as.data.frame(data\_core[data\_core$Abundance >0,])

rename\_core <- rename(dataset\_core\_filter, category = core)

dataset\_indicator\_filter <- as.data.frame(data\_indicator[data\_indicator$Abundance >0,])

rename\_indicator <- rename(dataset\_indicator\_filter, category = indicator)

binddataset <- rbind(rename\_Specific, rename\_core, rename\_indicator)

set.seed(3467)

ggplot(binddataset, aes(x = Abundance, y = fct\_reorder(Genus, Abundance, .desc = F), fill=category)) +

geom\_density\_ridges(scale = 1, alpha=0.8) +

#scale\_fill\_cyclical(values = c("blue", "green","red"))+

theme\_bw() + scale\_x\_continuous(trans="log10") + scale\_fill\_brewer(palette = "Dark2")

#图4a

install.packages(c("ggplot2", "dplyr", "scales", "reshape2", "tibble", "tidyverse",

"rstatix", "broom", "nortest", "vegan", "geosphere", "fANCOVA",

"phyloseqCompanion", "performance", "ape", "readxl"))

install.packages("BiocManager")

BiocManager::install(c("phyloseq", "qiime2R", "rbiom"))

devtools::install\_github("fishualize/fishualize")

devtools::install\_github("speedyseq/speedyseq")

devtools::install\_github("statnet/fANCOVA")

library(speedyseq)

library(phyloseq)

library(phyloseqCompanion)

library(geosphere)

library(vegan)

library(rbiom)

library(scales)

library(ggplot2)

library(dplyr)

library(fishualize)

library(ggpubr)

library(reshape2)

library(performance)

library(fANCOVA)

library(tibble)

library(qiime2R)

library(ape)

library(readxl)

library(tidyverse)

library(rstatix)

library(broom)

library(nortest)

#读取数据

bacteria\_ASV <- read.csv(file = "D:/study/master/meiji/bacteria\_ASV.csv",

sep=",",header=TRUE,check.names = FALSE)

# 定义文件的绝对路径（使用正斜杠）

rb\_metadata <- "D:/study/master/rb\_metadata.tsv"

# 导入元数据文件

rb\_metadata <- import\_qiime\_sample\_data(rb\_metadata)

# 查看元数据

head(rb\_metadata)

# 读取 ASV 表（特征表）

rb\_ASV <- bacteria\_ASV[, c("ASV",rb\_metadata$Sample.ID)]

rb\_ASV <- rb\_ASV %>% column\_to\_rownames(var = "ASV")

head(rb\_ASV)

# 读取 taxonomy（分类信息）

rb\_tax <- bacteria\_ASV[, 2:9]

rb\_tax <- rb\_tax %>% column\_to\_rownames(var = "ASV")

head(rb\_tax)

#分步生成和读取系统发育树

tree <- read\_qza("D:/study/master/meiji/rooted-tree.qza")

phylo\_tree <- tree$data

md5<-read\_excel("D:/study/master/meiji/ASV\_md5.xlsx")#md5值

rename\_vector <- setNames(md5$`ASV ID`, md5$md5)

phylo\_tree$tip.label <- rename\_vector[phylo\_tree$tip.label] #替换tip.label

# 合并为 phyloseq 对象

rb\_ASV <- otu\_table(rb\_ASV, taxa\_are\_rows = TRUE)

rb\_tax <- tax\_table(as.matrix(rb\_tax))

rb\_merged <- merge\_phyloseq(rb\_ASV, rb\_tax, rb\_metadata, phy\_tree(phylo\_tree))

#去除非细菌

rb\_merged <- phyloseq::subset\_taxa(rb\_merged, (Kingdom!="d\_\_Eukaryota") | is.na(Kingdom))

rb\_merged <- phyloseq::subset\_taxa(rb\_merged, (Kingdom!="d\_\_Archaea") | is.na(Kingdom))

rb\_merged <- phyloseq::subset\_taxa(rb\_merged, (Order!="o\_\_Chloroplast") )

rb\_merged <- phyloseq::subset\_taxa(rb\_merged, (Family!="f\_\_Mitochondria"))

rb\_merged <- phyloseq::subset\_taxa(rb\_merged, (Family!="NA"))

# 提取 ASV 表

rb\_ASV <- as.matrix(otu\_table(rb\_merged, taxa\_are\_rows=T))

rb\_ASV\_df <- rb\_ASV[rowSums(rb\_ASV[])>0,]

vegan\_otu <- function(physeq){

OTU <- otu\_table(physeq)

if(taxa\_are\_rows(OTU)){

OTU <- t(OTU)

}

return(as(OTU, "matrix"))

}

NOMIS\_df <- data.frame(rb\_metadata$longitude, rb\_metadata$latitude)#经纬度

NOMIS\_df$rb\_metadata.longitude <- as.numeric(NOMIS\_df$rb\_metadata.longitude)

NOMIS\_df$rb\_metadata.latitude <- as.numeric(NOMIS\_df$rb\_metadata.latitude)

#计算经纬度的地理距离矩阵

dist\_geo\_all <- distm(NOMIS\_df, NOMIS\_df, fun=distGeo)

dist\_geo\_all <- as.matrix(dist\_geo\_all)

diag(dist\_geo\_all)=NA

dist\_geo\_all\_diag <- t(matrix(t(dist\_geo\_all)[which(!is.na(dist\_geo\_all))],nrow=23,ncol=24))

min(dist\_geo\_all\_diag)

#计算物种丰度的Bray-Curtis相似性指数

vegan\_matrix\_all <-vegan\_otu(rb\_merged)

allregion\_bray <-vegdist(log1p(vegan\_matrix\_all), method="bray")# Bray-Curtis 距离矩阵

allregion\_m <- as.matrix(allregion\_bray)

diag(allregion\_m)=NA

allregion\_diag <-t(matrix(t(allregion\_m)[which(!is.na(allregion\_m))],nrow=23,ncol=24))

min(allregion\_diag)

#Mantel检验二者相关性

mantel(allregion\_diag, dist\_geo\_all\_diag, method = "pearson", permutations = 999, na.rm = TRUE)

dist\_all\_bray <- data.frame(BC\_dist\_bc=as.vector(allregion\_diag), BC\_sim\_bc=as.vector(1-allregion\_diag), geo\_dist=as.vector(dist\_geo\_all\_diag), Method="All rhizosphere")

ggplot(dist\_all\_bray, aes(x=geo\_dist/1000, y=BC\_sim\_bc)) +

geom\_point(size=1.2) + ylim(0,1) + xlim (0,20000)+

# facet\_wrap(~Method) +

stat\_smooth(method="lm", formula=y ~ (x), size=1.2, se=FALSE, linetype="solid") +

ylab("Community Similarity - Bray-Curtis") +

xlab("Geographic distance (km)") + ggtitle("") + theme\_bw() +

theme(strip.text.x=element\_text(size=10, color="black", face="bold.italic")) +

theme(strip.background=element\_rect(colour="black", fill="white")) +

theme(axis.text=element\_text(size=14),

axis.title=element\_text(size=14),

legend.text=element\_text(size=14)) +

theme(panel.grid.major = element\_blank(), panel.grid.minor = element\_blank(),

panel.background = element\_blank(), axis.line = element\_line(colour = "black"))+

scale\_x\_log10()

dist\_all\_bray %>% #回归模型的截距和斜率

do({

mod = lm(BC\_sim\_bc ~ geo\_dist, data = dist\_all\_bray)

data.frame(Intercept = coef(mod)[1],

Slope = coef(mod)[2])

})

asv\_table\_unif<-otu\_table(rb\_merged, taxa\_are\_rows=T)

unifrac\_essai<-phyloseq::UniFrac(rb\_merged, weighted=T, normalized=TRUE, parallel=FALSE, fast=TRUE)# 加权 UniFrac（考虑物种丰度信息）距离矩阵

allregion\_unif<-as.matrix(unifrac\_essai)

diag(allregion\_unif)=NA

allregion\_diag\_unif<-t(matrix(t(allregion\_unif)[which(!is.na(allregion\_unif))],nrow=23,ncol=24))

mantel(allregion\_diag\_unif, dist\_geo\_all\_diag, method = "pearson", permutations = 999, na.rm = TRUE)

dist\_all\_wunif <- data.frame(dis\_wunif=as.vector(allregion\_diag\_unif), sim\_wunif=as.vector(1-allregion\_diag\_unif), geo\_dist=as.vector(dist\_geo\_all\_diag), Method="All rhizosphere")

ggplot(dist\_all\_wunif, aes(x=geo\_dist/1000, y=sim\_wunif)) +

geom\_point(size=1.2) + ylim(0,1) + xlim (0,20000)+

# facet\_wrap(~Method) +

stat\_smooth(method="lm", formula= y ~ (x), size=1.2, se=FALSE, linetype="solid") +

ylab("Community Similarity - Weighted Unifrac") +

xlab("Geographic distance (km)") + ggtitle("") + theme\_bw() +

theme(strip.text.x=element\_text(size=10, color="black", face="bold.italic")) +

theme(strip.background=element\_rect(colour="black", fill="white")) +

theme(axis.text=element\_text(size=14),

axis.title=element\_text(size=14),

legend.text=element\_text(size=14)) +

theme(panel.grid.major = element\_blank(), panel.grid.minor = element\_blank(),

panel.background = element\_blank(), axis.line = element\_line(colour = "black"))+

scale\_x\_log10()

dist\_all\_wunif %>%

do({

mod = lm(sim\_wunif ~ geo\_dist, data = .)

data.frame(Intercept = coef(mod)[1],#截距

Slope = coef(mod)[2])#斜率

})

asv\_table\_unif <-otu\_table(rb\_merged, taxa\_are\_rows=T)

uwunifrac\_essai<-phyloseq::UniFrac(rb\_merged, weighted=F, normalized=TRUE, parallel=FALSE, fast=TRUE)#无加权

allregion\_uwnif<-as.matrix(uwunifrac\_essai)

diag(allregion\_uwnif)=NA

allregion\_diag\_uwnif<-t(matrix(t(allregion\_uwnif)[which(!is.na(allregion\_uwnif))],nrow=23,ncol=24))

mantel(allregion\_diag\_uwnif, dist\_geo\_all\_diag, method = "pearson", permutations = 999, na.rm = TRUE)

dist\_all\_uw <-data.frame(dis\_uw=as.vector(allregion\_diag\_uwnif), sim\_uw=as.vector(1-allregion\_diag\_uwnif), geo\_dist=as.vector(dist\_geo\_all\_diag), Method="All rhizosphere")

ggplot(dist\_all\_uw, aes(x=geo\_dist/1000, y=sim\_uw)) +

geom\_point(size=1.2) + ylim(0,1) + xlim (0,20000)+

# facet\_wrap(~Method) +

stat\_smooth(method="lm", formula= y ~ (x), size=1.2, se=FALSE, linetype="solid") +

ylab("Community Similarity - UnWeighted Unifrac") +

xlab("Geographic distance (km)") + ggtitle("") + theme\_bw() +

theme(strip.text.x=element\_text(size=10, color="black", face="bold.italic")) +

theme(strip.background=element\_rect(colour="black", fill="white")) +

theme(axis.text=element\_text(size=14),

axis.title=element\_text(size=14),

legend.text=element\_text(size=14)) +

theme(panel.grid.major = element\_blank(), panel.grid.minor = element\_blank(),

panel.background = element\_blank(), axis.line = element\_line(colour = "black"))+

scale\_x\_log10()

dist\_all\_uw %>%

do({

mod = lm(sim\_uw ~ geo\_dist, data = .)

data.frame(Intercept = coef(mod)[1],

Slope = coef(mod)[2])

})

vegan\_matrix\_all <-vegan\_otu(rb\_merged)

allregion\_sor <-vegdist(vegan\_matrix\_all, binary=T)#使用 SORENSEN 进行全距离衰减

allregion\_sor\_m <-as.matrix(allregion\_sor)

diag(allregion\_sor\_m)=NA

allregion\_sor\_diag <-t(matrix(t(allregion\_sor\_m)[which(!is.na(allregion\_sor\_m))],nrow=23,ncol=24))

min(allregion\_sor\_diag)

mantel(allregion\_sor\_diag, dist\_geo\_all\_diag, method = "pearson", permutations = 999, na.rm = TRUE)

dist\_all\_sor <-data.frame(dis\_sor=as.vector(allregion\_sor\_diag), sim\_sor=as.vector(1-allregion\_sor\_diag), geo\_dist=as.vector(dist\_geo\_all\_diag), Method="All rhizosphere")

ggplot(dist\_all\_sor, aes(x=geo\_dist/1000, y=sim\_sor)) +

geom\_point(size=1.2) + ylim(0,1) + xlim (0,20000)+

stat\_smooth(method="lm", formula=y ~ (x), size=1.2, se=FALSE, linetype="solid") +

ylab("Community Similarity - Sorensen") +

xlab("Geographic distance (km)") + ggtitle("") + theme\_bw() +

theme(strip.text.x=element\_text(size=10, color="black", face="bold.italic")) +

theme(strip.background=element\_rect(colour="black", fill="white")) +

theme(axis.text=element\_text(size=14),

axis.title=element\_text(size=14),

legend.text=element\_text(size=14)) +

theme(panel.grid.major = element\_blank(), panel.grid.minor = element\_blank(),

panel.background = element\_blank(), axis.line = element\_line(colour = "black"))+

scale\_x\_log10()

dist\_all\_sor %>%

do({

mod = lm(sim\_sor ~ geo\_dist, data = .)

data.frame(Intercept = coef(mod)[1],

Slope = coef(mod)[2])

})

melt\_bc <- melt(allregion\_diag)#组合不同衰减模式

melt\_bc$dis <- "BC"

melt\_sor <- melt(allregion\_sor\_diag)

melt\_sor$dis <- "SOR"

melt\_geo <- melt(dist\_geo\_all\_diag)

colnames(melt\_geo) <- c("Var1","Var2","dist\_geo")

bind\_bcsor <- rbind(melt\_bc, melt\_sor)

merge\_dis\_geo <- merge(bind\_bcsor, melt\_geo)

bcsor\_plot<- ggplot(merge\_dis\_geo) +

geom\_point(aes(y=1-value, x = dist\_geo, color= dis)) +

stat\_smooth(aes(y=1-value, x = dist\_geo, color=dis), method="lm", formula=y ~ (x), size=1.2, se=FALSE, linetype="solid") +

scale\_color\_fish(option="Scarus\_quoyi",discrete=T)+

ylab("Community Similarity") +

xlab("Geographic distance (km)") + ggtitle("") + theme\_bw() +

scale\_x\_log10() +theme\_bw() + theme(panel.border = element\_blank(), panel.grid.major = element\_blank(),

panel.grid.minor = element\_blank(), axis.line = element\_line(colour = "black"))

bcsor\_plot

melt\_unifrac <- melt(allregion\_diag\_unif)#组合有加权和无加权

melt\_unifrac$dis <- "wunifrac"

melt\_uwunifrac <- melt(allregion\_diag\_uwnif)

melt\_uwunifrac$dis <- "unwunifrac"

bind\_unifrac<- rbind(melt\_unifrac, melt\_uwunifrac)

merge\_dis\_unif <- merge(bind\_unifrac, melt\_geo)

unifrac\_plot<- ggplot(merge\_dis\_unif) +

geom\_point(aes(y=1-value, x = dist\_geo, color= dis)) +

stat\_smooth(aes(y=1-value, x = dist\_geo, color=dis), method="lm", formula=y ~ (x), size=1.2, se=FALSE, linetype="solid") +

scale\_color\_fish(option="Trimma\_lantana",discrete=T)+

ylab("Community Similarity") +

xlab("Geographic distance (km)") + ggtitle("") + theme\_bw() +

scale\_x\_log10() +theme\_bw() + theme(panel.border = element\_blank(), panel.grid.major = element\_blank(),

panel.grid.minor = element\_blank(), axis.line = element\_line(colour = "black"))

unifrac\_plot

bind\_all\_dist <- rbind(melt\_bc, melt\_sor, melt\_uwunifrac, melt\_unifrac)#组合所有距离矩阵

merge\_alldist\_geo <- merge(bind\_all\_dist, melt\_geo)

get\_fishcol <- fish(4, option="Trimma\_lantana")#颜色

alldist\_plot<- ggplot(merge\_alldist\_geo ) +

geom\_point(aes(y=1-value, x = dist\_geo/1000, color= dis), alpha=0.1) +

stat\_smooth(aes(y=1-value, x = dist\_geo/1000, color=dis), method="lm", formula=y ~ (x), size=1.2, se=FALSE, linetype="solid") +

scale\_color\_fish(option="Scarus\_quoyi",discrete=T,direction=-1)+

ylab("Community Similarity") +

xlab("Geographic distance (km)") + ggtitle("") + theme\_bw() +

scale\_x\_log10() +theme\_bw() + theme(panel.border = element\_blank(), panel.grid.major = element\_blank(),

panel.grid.minor = element\_blank(), axis.line = element\_line(colour = "black"))

alldist\_plot

merge\_alldist\_geo$dist\_geo <- as.numeric(merge\_alldist\_geo$dist\_geo)

merge\_alldist\_geo$dist\_geo\_km <- merge\_alldist\_geo$dist\_geo / 1000#地理距离转换为公里

calculate\_r\_squared <- function(model) {#计算r平方

return(format(summary(model)$r.squared, digits = 3))

}

lm\_models\_tax <- by(merge\_alldist\_geo, merge\_alldist\_geo$dis, function(sub\_df\_tax) {

lm(1 - sub\_df\_tax$value ~ sub\_df\_tax$dist\_geo\_km, data = sub\_df\_tax)

})#线性回归，群落相似性随地理距离的变化趋势

intercepts <- sapply(lm\_models\_tax, function(model) coef(model)[1])#计算r平方等

slopes <- sapply(lm\_models\_tax, function(model) coef(model)[2])

r\_squared <- sapply(lm\_models\_tax, calculate\_r\_squared)

#subset\_size <- 35000#随机抽样

#set.seed(42)

#subset\_merge\_alldist <- merge\_alldist\_geo[sample(nrow(merge\_alldist\_geo), subset\_size), , drop = FALSE]

subset\_merge\_alldist <- merge\_alldist\_geo

#差异性优势模式比较，主要优势菌群

merge\_bcsor <- merge(dist\_all\_bray, dist\_all\_sor, by="row.names") #Bray-Curtis vs Sorensen

anco\_bcsor <- merge\_bcsor[c("BC\_sim\_bc","sim\_sor","geo\_dist.x")]

manco <- melt(anco\_bcsor, id=c("geo\_dist.x"))

ggscatter(

manco, x = "geo\_dist.x", y = "value",

color = "variable", add = "reg.line"

)+

stat\_regline\_equation(

aes(label = paste(..eq.label.., ..rr.label.., sep = "~~~~"), color = variable)

)

manco$geo\_dist\_log <- log((manco$geo\_dist.x)/1000)

merge\_unifrac <- merge(dist\_all\_wunif, dist\_all\_uw, by="row.names")#Weighted vs Unweighted unifrac

anco\_unifrac <- merge\_unifrac[c("sim\_wunif","sim\_uw","geo\_dist.x")]

manco\_unifrac <- melt(anco\_unifrac, id=c("geo\_dist.x"))

manco\_unifrac$geo\_dist\_log <- log((manco\_unifrac$geo\_dist.x)/1000)

ggscatter(

manco\_unifrac, x = "geo\_dist.x", y = "value",

color = "variable", add = "reg.line"

)+

stat\_regline\_equation(

aes(label = paste(..eq.label.., ..rr.label.., sep = "~~~~"), color = variable)

)

manco\_unifrac$geo\_dist\_log <- log((manco\_unifrac$geo\_dist.x)/1000)

manco\_clean <- manco %>% filter(geo\_dist\_log >= 0)

manco\_clean %>% anova\_test(sqrt(value) ~ variable \* geo\_dist\_log)

model\_bcsor <- lm((value) ~ geo\_dist\_log + variable, data = manco\_clean)# 残差的正态性，拟合模型，协变量优先

model\_metrics\_bcsor <- augment(model\_bcsor) %>%# 检查模型诊断指标

select(-.hat, -.sigma, -.fitted) #减少细节

head(model\_metrics\_bcsor, 3)

ad.test(model\_bcsor$residuals) #Anderson-Darling 正态性检验

manco\_unifrac %>% anova\_test(log10(value) ~variable\*geo\_dist.x)# 回归斜率的同质性

manco\_clean$geo\_dist\_log\_centered <- datawizard::standardize(manco\_clean$geo\_dist\_log, center = TRUE, scale = FALSE) # 标准化变量

glm\_dd\_bcsor <- lm(sqrt(value) ~ geo\_dist\_log\_centered \* variable, data = manco\_clean) # 用中心变量拟合模型

check\_model(glm\_dd\_bcsor)

summary(glm\_dd\_bcsor)

manco\_unifrac\_clean <- manco\_unifrac %>% filter(geo\_dist\_log >= 0)

manco\_unifrac\_clean$geo\_dist\_log <- log((manco\_unifrac\_clean$geo\_dist.x)/1000)

manco\_unifrac\_clean$geo\_dist\_log\_centered <- datawizard::standardize(manco\_unifrac\_clean$geo\_dist\_log, center = TRUE, scale = FALSE)

glm\_dd\_wufrac <- lm(manco\_unifrac\_clean, formula = value ~ geo\_dist\_log\_centered\*variable)

check\_model(glm\_dd\_wufrac)

summary(glm\_dd\_wufrac)

#图4b

#图5a

if (!require("BiocManager", quietly = TRUE))

install.packages("BiocManager")

BiocManager::install("ggtree")

BiocManager::install("ggtreeExtra")

library(tidyverse)

library(readr)

library(readxl)

library(data.table)

library(phyloseq)

library(qiime2R)

library(ape)

library(magrittr)

library(dplyr)

library(tidyr)

library(iCAMP)

library(ggtree)

library(ggtreeExtra)

library(phytools)

library(ggplot2)

library(ggnewscale)

library(castor)

library(viridis)

library(phangorn)

#读取数据

bacteria\_ASV <- read.csv(file = "D:/study/master/meiji/bacteria\_ASV.csv",

sep=",",header=TRUE,check.names = FALSE)

# 定义文件的绝对路径（使用正斜杠）

rb\_metadata <- "D:/study/master/rb\_metadata.tsv"

# 导入元数据文件

rb\_metadata <- import\_qiime\_sample\_data(rb\_metadata)

# 查看元数据

head(rb\_metadata)

# 读取 ASV 表（特征表）

rb\_ASV <- bacteria\_ASV[, c("ASV",rb\_metadata$Sample.ID)]

rb\_ASV <- rb\_ASV %>% column\_to\_rownames(var = "ASV")

head(rb\_ASV)

# 读取 taxonomy（分类信息）

rb\_tax <- bacteria\_ASV[, 2:9]

rb\_tax <- rb\_tax %>% column\_to\_rownames(var = "ASV")

head(rb\_tax)

#分步生成和读取系统发育树

tree <- read\_qza("D:/study/master/meiji/rooted-tree.qza")

phylo\_tree <- tree$data

md5<-read\_excel("D:/study/master/meiji/ASV\_md5.xlsx")#md5值

rename\_vector <- setNames(md5$`ASV ID`, md5$md5)

phylo\_tree$tip.label <- rename\_vector[phylo\_tree$tip.label] #替换tip.label

# 合并为 phyloseq 对象

rb\_ASV <- otu\_table(rb\_ASV, taxa\_are\_rows = TRUE)

rb\_tax <- tax\_table(as.matrix(rb\_tax))

rb\_merged <- merge\_phyloseq(rb\_ASV, rb\_tax, rb\_metadata, phy\_tree(phylo\_tree))

#去除非细菌

rb\_merged <- phyloseq::subset\_taxa(rb\_merged, (Kingdom!="d\_\_Eukaryota") | is.na(Kingdom))

rb\_merged <- phyloseq::subset\_taxa(rb\_merged, (Kingdom!="d\_\_Archaea") | is.na(Kingdom))

rb\_merged <- phyloseq::subset\_taxa(rb\_merged, (Order!="o\_\_Chloroplast") )

rb\_merged <- phyloseq::subset\_taxa(rb\_merged, (Family!="f\_\_Mitochondria"))

rb\_merged <- phyloseq::subset\_taxa(rb\_merged, (Family!="NA"))

# 提取 ASV 表

rb\_ASV <- as.matrix(otu\_table(rb\_merged, taxa\_are\_rows=T))

rb\_ASV\_df <- rb\_ASV[rowSums(rb\_ASV[])>0,]

# 提取组信息

groups <- unique(rb\_metadata$Group)

# 创建每个组的文件夹并拆分数据

for (group in groups) {

# 为每个组创建一个文件夹

group\_wd <- paste0("D:/study/master/iCAMP/", group)

if (!dir.exists(group\_wd)) {

dir.create(group\_wd, recursive = TRUE)

}

# 筛选该组的样本

group\_samples <- rownames(subset(rb\_metadata, Group == group)) # 获取属于该组的样本ID

# 根据样本筛选ASV表

group\_ASV <- rb\_ASV[, colnames(rb\_ASV) %in% group\_samples]

# 获取该组中存在的 ASV

group\_ASV\_ids <- rownames(group\_ASV[rowSums(group\_ASV) > 0, ]) # 只保留非零 ASV

# 根据 ASV 筛选分类信息

group\_tax <- rb\_tax[rownames(rb\_tax) %in% group\_ASV\_ids, ]

# 根据 ASV 筛选系统发育树

group\_tree <- drop.tip(phylo\_tree, setdiff(phylo\_tree$tip.label, group\_ASV\_ids)) # 移除不属于该组的 ASV

# 保存拆分后的数据

write.table(group\_ASV, file = paste0(group\_wd, "/Asv\_table.txt"), sep = "\t", quote = FALSE)

write.table(group\_tax, file = paste0(group\_wd, "/classification.txt"), sep = "\t", quote = FALSE)

write.tree(group\_tree, file = paste0(group\_wd, "/tree.nwk"))

}

# 获取所有组的文件夹

dirs <- list.dirs("D:/study/master/iCAMP", full.names = TRUE, recursive = FALSE)

# 遍历每个文件夹并执行iCAMP分析，分析群落构建过程，分解为不同的生态过程（如选择、扩散、漂变等）

for (group\_wd in dirs) {

# 设置文件路径

com.file <- paste0(group\_wd, "/Asv\_table.txt")

tree.file <- paste0(group\_wd, "/tree.nwk")

clas.file <- paste0(group\_wd, "/classification.txt")

# 设置输出文件夹

save.wd <- paste0(group\_wd, "/out2")

if (!dir.exists(save.wd)) {

dir.create(save.wd)

}

# 参数设置

prefix <- basename(group\_wd) # 使用组名作为输出文件的前缀

rand.time <- 1000

nworker <- 48

memory.G <- 200

# 读取数据

comm <- t(read.table(com.file, header = TRUE, sep = "\t", row.names = 1, check.names = FALSE))

tree <- read.tree(file = tree.file)

clas <- read.table(clas.file, header = TRUE, sep = "\t", row.names = 1, check.names = FALSE)

# 匹配OTU ID

spid.check <- match.name(cn.list = list(comm = comm), rn.list = list(clas = clas), tree.list = list(tree = tree))

comm <- spid.check$comm

clas <- spid.check$clas

tree <- spid.check$tree

# 计算系统发育距离矩阵

pd.big <- iCAMP::pdist.big(tree = tree, wd = save.wd, nworker = nworker, memory.G = memory.G)

# iCAMP分析

bin.size.limit <- 48

sig.index <- "Confidence"

#基于物种丰度和系统发育距离，分析群落构建过程

icres <- iCAMP::icamp.big(comm = comm, pd.desc = pd.big$pd.file, pd.spname = pd.big$tip.label,

pd.wd = pd.big$pd.wd, rand = rand.time, tree = tree, prefix = prefix,

ds = 0.2, pd.cut = NA, sp.check = TRUE,

phylo.rand.scale = "within.bin", taxa.rand.scale = "across.all",

phylo.metric = "both", sig.index = sig.index, bin.size.limit = bin.size.limit,

nworker = nworker, memory.G = memory.G, rtree.save = FALSE,

detail.save = TRUE, qp.save = FALSE, detail.null = FALSE,

ignore.zero = TRUE, output.wd = save.wd, correct.special = TRUE,

unit.sum = rowSums(comm), special.method = "depend",

ses.cut = 1.96, rc.cut = 0.95, conf.cut = 0.975, omit.option = "no", meta.ab = NULL)

# 保存结果

bins <- icres$detail$taxabin$sp.bin

res1 <- icres$CbMPDiCBraya

write.csv(res1, paste0(group\_wd, "/", prefix, ".res.csv"))

write.csv(bins, paste0(group\_wd, "/", prefix, ".bins.csv"))

#对结果进行分组统计，进一步解析不同系统发育 bin 的生态过程

icbin <- iCAMP::icamp.bins(icamp.detail = icres$detail, clas = clas, silent = FALSE, boot = FALSE,

rand.time = rand.time, between.group = FALSE)

save(icbin, file = paste0(group\_wd, "/", prefix, ".iCAMP.Summary.rda"))

write.csv(icbin$Pt, file = paste0(group\_wd, "/", prefix, ".ProcessImportance\_EachGroup.csv"), row.names = FALSE)

write.csv(icbin$Ptk, file = paste0(group\_wd, "/", prefix, ".ProcessImportance\_EachBin\_EachGroup.csv"), row.names = FALSE)

write.csv(icbin$Ptuv, file = paste0(group\_wd, "/", prefix, ".ProcessImportance\_EachTurnover.csv"), row.names = FALSE)

write.csv(icbin$BPtk, file = paste0(group\_wd, "/", prefix, ".BinContributeToProcess\_EachGroup.csv"), row.names = FALSE)

write.csv(data.frame(ID = rownames(icbin$Class.Bin), icbin$Class.Bin, stringsAsFactors = FALSE),

file = paste0(group\_wd, "/", prefix, ".Taxon\_Bin.csv"), row.names = FALSE)

write.csv(icbin$Bin.TopClass, file = paste0(group\_wd, "/", prefix, ".Bin\_TopTaxon.csv"), row.names = FALSE)

}

#组合ProcessImportance\_EachBin\_EachGroup.csv和Taxon\_Bin.csv，一般选择 CbMNTDiCbraya

iCAMP\_all\_combined <- data.frame()

for (group\_wd in dirs) {

prefix <- basename(group\_wd)

taxon\_bin\_data <- read.csv(paste0(group\_wd, "/", prefix, ".Taxon\_Bin.csv"), header=TRUE , stringsAsFactors = FALSE)

mntd\_data <- read.csv(paste0(group\_wd, "/", prefix, ".ProcessImportance\_EachBin\_EachGroup.csv"), header=FALSE , stringsAsFactors = FALSE)

taxon\_bin\_data <- taxon\_bin\_data[, c(1:3)]

mntd\_data <- as.data.frame(t(mntd\_data[, -c(1:3)]))

mntd\_data<-mntd\_data[,c(1,14)]

colnames(mntd\_data) <- as.character(mntd\_data[1, ])

mntd\_data <- mntd\_data[-1, ]

colnames(mntd\_data) <- c("Bin", "process")

mntd\_data$Bin <- gsub("bin", "Bin", mntd\_data$Bin)

# 合并 taxon\_bin\_data 和 mntd\_data，按 Bin 匹配

iCAMP\_data <- merge(taxon\_bin\_data, mntd\_data, by = "Bin", all.x = TRUE)

# 添加 group 列

iCAMP\_data$group <- prefix

# 合并到总数据框

iCAMP\_all\_combined <- rbind(iCAMP\_all\_combined, iCAMP\_data)

}

# 保存结果

write.table(iCAMP\_all\_combined, file = "D:/study/master/iCAMP/iCAMP\_all\_combined.csv",

sep = ",", row.names = FALSE, quote = FALSE)

ASVs <- as.data.frame(rb\_ASV\_df)

tree <- phy\_tree(rb\_merged)

tree <- midpoint(tree)# 树的中点根化

#平均相对丰度（外环）

mean.RA <- data.frame(rowMeans(ASVs))

mean.RA$ASVs <- rownames(ASVs)

mean.RA$log.RA <- log1p(mean.RA$rowMeans.ASVs.)

colnames(mean.RA) <- c("mean.RA","ASVs", "log.RA")

#显示每个分组的HoS、DL和DR的热图

all.ASVs <- read.csv("D:/study/master/iCAMP/iCAMP\_all\_combined.csv")

all.ASVs <- all.ASVs[all.ASVs$process =="DL" | all.ASVs$process == "HoS" | all.ASVs$process == "DR",]

tax <- as.data.frame(rb\_tax)

#选择的门（HoS 和/或 DL 在多个区域中重要）

p\_acid\_asvs <- tax[tax$Phylum==" p\_\_Acidobacteriota",]

p\_actino\_asvs <- tax[tax$Phylum==" p\_\_Actinobacteriota",]

p\_bact\_asvs <- tax[tax$Phylum==" p\_\_Bacteroidota",]

p\_bdello\_asvs <- tax[tax$Phylum==" p\_\_Bdellovibrionota",]

p\_gemma\_asvs <- tax[tax$Phylum==" p\_\_Gemmatimonadota",]

p\_latesc\_asvs <- tax[tax$Phylum==" p\_\_Latescibacterota",]

p\_myxo\_asvs <- tax[tax$Phylum==" p\_\_Myxococcota",]

p\_plan\_asvs <- tax[tax$Phylum==" p\_\_Planctomycetota",]

p\_prot\_asvs <- tax[tax$Phylum==" p\_\_Proteobacteria",]

p\_verr\_asvs <- tax[tax$Phylum==" p\_\_Verrucomicrobiota",]

taxa\_df <- data.frame(ASV=tree$tip.label, Family=rep('Others', length(tree$tip.label)), Phylum=rep('Others', length(tree$tip.label)))

taxa\_df$Phylum[taxa\_df$ASV %in% rownames(p\_acid\_asvs)] <- 'Acidobacteriota'

taxa\_df$Phylum[taxa\_df$ASV %in% rownames(p\_actino\_asvs)] <- 'Actinobacteriota'

taxa\_df$Phylum[taxa\_df$ASV %in% rownames(p\_bact\_asvs)] <- 'Bacteroidota'

taxa\_df$Phylum[taxa\_df$ASV %in% rownames(p\_bdello\_asvs)] <- 'Bdellovibrionota'

taxa\_df$Phylum[taxa\_df$ASV %in% rownames(p\_gemma\_asvs)] <- 'Gemmatimonadota'

taxa\_df$Phylum[taxa\_df$ASV %in% rownames(p\_latesc\_asvs)] <- 'Latescibacterota'

taxa\_df$Phylum[taxa\_df$ASV %in% rownames(p\_myxo\_asvs)] <- 'Myxococcota'

taxa\_df$Phylum[taxa\_df$ASV %in% rownames(p\_plan\_asvs)] <- 'Planctomycetota'

taxa\_df$Phylum[taxa\_df$ASV %in% rownames(p\_prot\_asvs)] <- 'Proteobacteria'

taxa\_df$Phylum[taxa\_df$ASV %in% rownames(p\_verr\_asvs)] <- 'Verrucomicrobiota'

# 绘制带有分类学、热图和平均相对丰度的树

p = ggtree(tree, layout="fan", size=0.25, open.angle=10)

p2<-p +

geom\_fruit(data=taxa\_df, geom=geom\_tile,

mapping=aes(y=ASV, fill=Phylum), width = 0.5,

offset = 0.02) + scale\_fill\_manual(values = c('#FA907B','#7bbbfa', '#C42D50','#46CF6B','#529C7E','#84CCFA', '#3234D9', 'white','#fafa7b','#bbfa7b','#cf46aa')) + new\_scale\_fill() +

geom\_fruit(

data=all.ASVs,

geom=geom\_tile,

mapping=aes(y=ID, x=group, fill=process),

offset=0.08, # The distance between external layers, default is 0.03 times of x range of tree.

pwidth=0.25 # width of the external layer, default is 0.2 times of x range of tree.

) +

scale\_fill\_manual(

values=c("#3303df", "#dfac03","#df3e03"),

guide=guide\_legend(keywidth=0.5, keyheight=0.5, order=3)

)

p3<-p2 +

new\_scale\_fill()+

geom\_fruit(data=mean.RA,

geom=geom\_bar,

mapping=aes(y=ASVs, x=log.RA),

offset = 0.05, orientation='y',

stat="identity")

#ggsave(p3, filename = 'Figure\_phy\_tree.pdf', width = 15, height = 15)