Tree tissues and species traits modulate the microbial methane-cycling communities of the tree phyllosphere

Additional file 2: Q2Pipe Option file, and Script for statistical analyses

Marie-Ange Moisan[1,2], Vincent Maire[1,2], Marie-Josée Morency[3], Christine Martineau[3]

[1]Département des Sciences de l'environnement, Université du Québec à Trois-Rivières, 3351 Bd des Forges, Trois-Rivières, QC G8Z 4M3, Canada.

[2]Centre de Recherche sur les Interactions Bassins Versants - Écosystèmes Aquatiques (RIVE), Université du Québec à Trois-Rivières, 3351 Bd des Forges, Trois-Rivières, QC G8Z 4M3, Canada.

[3]Canadian Forest Service, Natural Resources Canada, Laurentian Forestry Centre, 1055 Rue du Peps, Québec, QC G1V 4C7, Canada.

1. Q2Pipe Option file

```
# Designed for Q2Pipe V0.95 #
#  Common Analysis Parameters
NB_THREADS=12
# Threads fine-tuning
CLASSIFIER_NB_THREADS=2

############################
#  Step 1: Data Import #
############################

# PLACEHOLDER for single end support DO NOT CHANGE THIS, ONLY PAIRED IS SUPPORTED
DATA_TYPE=paired

# Number of sequences to use to generate quality plots (default: 10000)
p_n=10000

### FALCO ###

# Launch Falco analysis on samples (similar to fastqc but faster implementation)
# This program will NEVER affect your reads, it is using reads copies
RUN_FALCO=true

# Trim the end of the reads to reduce false positive on Falco FAIL detection
# Use 0 to analyse the complete sequence (not recommended as you many of your sample will
fail
# screening because of the low-quality ends whit Illumina data
# Use 0 to disable the trimming
falco_right_trim=0

# Create a combined R1 file and a combined R2 file and run it through Falco
# Should give similar results to Qiime2 import, but could help flag entire run problems
# Will take more time to run and take more disk space
FALCO_COMBINED_RUN=true

# If true, will delete the exportation file to save disk space (results will be kept)
CLEAN_FALCO_OUTPUT=true

### FIGARO ###

# Launch Figaro program to predict correct trimming positions
# This program will NEVER affect your reads, it is using reads copies.
RUN_FIGARO=true

# Will substract the longest sequence length by this number
# This will make sure every reads are the same length
```

```
# NOTE : This will NOT affect your reads, only the cpies
figaro_trim_offset=2

# size must EXCLUDE the primers, ff amplicon length is variable, use the longest length expected
# If you want to test different amplicon size, you can input multiple values (separate them by ,)
# NOTE: each amplicon size takes approximately 40s
f_amplicon_size=368,370,372,374,376,378,380
f_forward_primer_len=19
f_reverse_primer_len=20

# 12 is default value in Qiime2
f_min_overlap=12

# If true, will delete the exportation file to save disk space (results will be kept)
CLEAN_FIGARO_OUTPUT=true

##########################
#  Cutadapt #
##########################
# This step is done during step 1.5 and step 2

RUN_CUTADAPT=false
forward_trim_param=--p-front-f
reverse_trim_param=--p-front-r

forward_primer=GTGYCAGCMGCCGCGGTAA
reverse_primer=CCGYCAATTYMTTTRAGTTT

p_discard_untrimmed=true

# Ratio of error in the adapter recognition
# The actual error rate is computed as the number of errors in the match
# divided by the length of the matching part of the adapter
# Ex: an adapter match of length 8 containing 1 error has an error rate of 1/8=0.125
p_error_rate=0.1

# if true, will force the step to stop just after cutadapt
# Useful for debugging and accessing CutAdapt qzv files
CA_FORCE_INTERRUPTION=false

#########################################################
#  Step 1.5: Dada2 Denoising Parameters Evaluation #
#########################################################

# This step is design to work with a single manifest file
# Manifest must be in CSV format
EVAL_MANIFEST_FILE_PATH=
```

```
# if your data is single-end, only p_trim_left_f, p_trunc_len_f and p_max_ee_f will be used in trimming

# Use 0 to bypass evaluation (will just denoise all samples)
DENOISE_EVALUATION_SAMPLE_SIZE=5
TESTFILE_PATH=
# if true, will generate a new random manifest. even if one already exists
# if false, will use the $ANALYSIS_NAME.eval_manifest.temp as subset
FORCE_RESAMPLING=false

# If true, will stop the step after generating $ANALYSIS_NAME.eval_manifest.temp
# Nothing else will be generated until DRY_RUN is switched to false again
DRY_RUN=false

#################################
# Step 2: Dada2 Denoising #
#################################

# NOT FOR STEP 1.5 (YOU MUST USE TESTFILE FOR STEP 1.5)

# Number of denoising job to launch simultaneously
# Each job will use NB_THREADS / job count
# For best result, your number of job should be a multiple of your NB_THREADS
# Ex: 3 jobs should have 9, 12, 15, etc. NB_THREADS so each job have the same number of threads
# If 1 is used, all job will run sequentially (which will be slower that having multiple jobs)
CONCURRENT_JOBS=4

# Use 0 to deactivate parameters (should be 0 if you used CutAdapt)
p_trim_left_f=19
p_trim_left_r=20

# Use 0 to deactivate parameters
# If you have multiple run, you must input a value for each of them
# seperate them by a comma (ex 234,235,223)
# They must be in the same order than your manifest list
p_trunc_len_f=262,262,262,262
p_trunc_len_r=200,200,200,200

p_max_ee_f=2
p_max_ee_r=2
p_n_reads_learn=1000000
p_trunc_q=2

p_chimera_method=consensus
p_min_fold_parent_over_abundance=1.0
```

```
############################
# Step 2.5 : Run Merging  #
############################

# If true, will proceed with merging even if runs are incompatible (different denoising
parameters)
# NOT RECOMMENDED
IGNORE_INCOMPATIBILITY=false

#####################################
# Step 3: Feature Table filtering #
#####################################

# FUTURE DEV: REPLACE OR ADD AUTO CALCULATION ACCORDING TO VISUALIZATION DATA
# Use 0 to disable
#freq_threshold=0.0005
p_min_frequency=0

# IDEM ^^^^
# Use 0 to disable
#sample_threshold=0
p_min_samples=0

################################
# Step 4: Denovo Clustering  #
################################

# Must be between ]0 , 1.0]
# Use NA to skip denovo clustering (you must still launch step 4)
p_perc_identity=NA

#################################
# Step 5: Classifier Training   #
#################################

# Only Specify these if you want to train the classifier with your database
# It's recommended to use the same name (different extension) for the fasta and the tax file
# The classifier output name must have the .qza extension

#FASTA_DATABASE_PATH=
#TAXO_DATABASE_PATH=
SEQS_QZA_PATH=
TAXO_QZA_PATH=
CLASSIFIER_OUTPUT_NAME=

################################
# Step 6 : Taxonomy Assignment  #
################################
```

```
# Only specify this if you skipped step 5
CLASSIFIER_DATABASE_PATH=/mnt/genomics/CMartineau/Qiime2_databases/16S_18S/silva_13
8_18S_Corrected/classifier_silva138_full_18SGcor.q2023_5.qza

# Possible value [0, 1]
p_confidence=0.7

############################
# Step 7: Taxa Filtering #
############################

# Filter samples according to specific metadata
# ex "[subject]='subject-1' AND NOT [body-site]='gut'"
# ex "[body-site]='gut' OR [reported-antibiotic-usage]='Yes'"
# ex "[body-site] IN ('left palm', 'right palm')"
# Leave "" to ignore
p_where=""

# Never use space in the list, only comma
# p_exclude remove unwanted taxe and p_include keep only specified taxa
# You can use both option to exclude parts of the inclusion
# If both are empty, taxa filtering will be skipped
p_include=bacteria,archaea
p_exclude=eukaryota,mitochondria,chloroplast

# Must be exact or contains (default)
p_mode=contains

# Will also be used in step 8 and 10
GENERATE_PHYLOGENY=true

# RAREFACTION OVERRIDE
# true, false or both
# if you activate this option, you can ignore the next sections, skip step 8 and 9
# Use "both" without the quotes to run both with and without rarefaction in a single run
SKIP_RAREFACTION=both
########################################
#   Step 8: Alpha rarefaction plotting #
########################################

p_max_depth=10000
p_steps=50
p_iterations=10
# Use this format: observed_features,shannon
# You can also leave this blank to use Qiime2 default metrics (observed_features,shannon)
# IMPORTANT: This is NOT the metrics for step 10, this is only for the rarefaction curve
p_metrics=observed_features,shannon
```

```
##############################
# Step 9 : Data Rarefaction #
##############################

p_sampling_depth=3300

########################
# Step 10 : Metrics #
########################

alpha_metrics=shannon,simpson,observed_features,chao1,pielou_e
beta_metrics=braycurtis,jaccard

# Relevent only if GENERATE_PHYLOGENY is activated
alpha_metrics_phylo=faith_pd
beta_metrics_phylo=weighted_unifrac,weighted_normalized_unifrac,generalized_unifrac,unwei
ghted_unifrac

######################
# Step 11: Export #
######################

## FUNGuild specific options (Fungal ITS Only)
# To use FUNGuild, Guilds_v1.1.py must be available in your path
# Recommended FUNGuild version : https://github.com/Patg13/FUNGuild
GENERATE_FUNGUILD=false
# Enter remote_fungi to use remote fungi database (http://stbates.org/funguild_db.php)
# You can also specify a local database path
# Database must be downloaded with the provided bash script in
https://github.com/Patg13/FUNGuild
FUNGUILD_DATABASE_PATH=

# Path to the result extraction form (in tsv format)
EXTRACTION_FORM_PATH=

## ANCOM specific options
GENERATE_ANCOM=false
# Collapse table at genus level (6)
p_level=6

# Depend on your metadata
# Specify each target column seperated by a comma
# Ex: Col1,Col2,Col3,Col4
# There will be one ANCOM by column
m_metadata_column=
```

2. Script for statistical analyses of microbial data

#These analyses are performed from the .biom files generated from the Q2Pipe bioinformatic processing of raw sequences

 # Loading librairies for analyses#

```
library(phyloseq)
library(Biostrings)
library(ggplot2)
library(vegan)
library(grid)
library(microbiome)
library(ggpubr)
library(AICcPermanova)
library(pairwiseAdonis)
library(tidyverse)
library(ANCOMBC)
library(emmeans)
library(AICcmodavg)
library(multcompView)
library(multcomp)
library(visreg)
```

##LOADING DATA INTO PHYLOSEQ OBJECTS##

#Load the biom file (generated from the rarefied and non-rarefied data) and metadata into a phyloseq object#

```
#Rarefied data #
ps <- read_biom2phyloseq(biom.file = "feature_taxonomy_merged.biom", metadata.file =
"meta_all.csv", taxonomy.file = NULL)
sample<-sample_data(ps)

#Non rarefied data #
ps_non_rar <- read_biom2phyloseq(biom.file = "feature_taxonomy_merged.biom", metadata.file =
"meta_all.csv", taxonomy.file = NULL)
samplenr<-sample_data(ps_non_rar)
```

#Create sub-objects for compartments and tissues (16S microbial communities) #

```
epi <- subset_samples(ps,Compartment== "Leaf_epiphyte") ##Change to ps_non_rar for generating the
objects from non rarefied data
endo <- subset_samples(ps,Compartment== "Leaf_endophyte")
bark <- subset_samples(ps,Compartment== "Bark")
hw <- subset_samples(ps,Compartment== "Heartwood")
sw <- subset_samples(ps,Compartment== "Sapwood")
```

```
wood <- subset_samples(ps,Tissue== "Wood")
leaf <- subset_samples(ps,Tissue== "Leaf")
```

#Create sub-objects of methanotrophic/methanogenic communities#

```
methanogen <- subset_taxa(ps, Class =='c__Methanobacteria'|Class =='c__Methanosarcinia'|Order
=='o__Methanomassiliicoccales'|Class =='c__Methanomicrobia'| Order =='o__Methanococcales'|Order
=='o__Methanopyrales'|Order =='Methanocellales'|Class =='Methanofastidiosa')

methanotroph <- subset_taxa(ps,
Genus=='g__Clonothrix'|Genus=='g__Crenothrix'|Genus=='g__Methylacidiphilum'|Genus=='g__Methylaci
dimicrobium'|Genus=='g__Methylobacter'|Genus=='g__Methylocaldum'|Genus=='g__Methylocapsa'|Ge
nus=='g__Methylocella'|Genus=='g__Methylococcus'|Genus=='g__Methylocystis'|Genus=='g__Methylofe
rula'|Genus=='g__Methylogaea'|Genus=='g__Methyloglobulus'|Genus=='g__Methylohalobius'|Genus=='
g__Methylomagnum'|Genus=='g__Methylomarinovum'|Genus=='g__Methylomarinum'|Genus=='g__Met
hylobacterium-
Methylorubrum'|Genus=='g__Methylorosula'|Genus=='g__Methylomonas'|Genus=='g__Methyloparacoc
cus'|Genus=='g__Methylosarcina'|Genus=='g__Methylosinus'|Genus=='g__Methylosoma'|Genus=='g__
Methylosphaera'|Genus=='g__Methylothermus'|Genus=='g__Methylomirabilis'|Genus=='g__Methyloce a
nibacter'|Genus=='g__Methylogellaceae'|Genus=='g__pLW-
20'|Genus=='g__Methyloligellaceae'|Genus=='g__Methylomicrobium')

methano <- subset_taxa(ps, Class =='c__Methanobacteria'|Class =='c__Methanosarcinia'|Order
=='o__Methanomassiliicoccales'|Class =='c__Methanomicrobia'|Order =='o__Methanococcales'|Order
=='o__Methanopyrales'|Order =='Methanocellales'|Class
=='Methanofastidiosa'|Genus=='g__Clonothrix'|Genus=='g__Crenothrix'|Genus=='g__Methylacidiphilum'
|Genus=='g__Methylacidimicrobium'|Genus=='g__Methylobacter'|Genus=='g__Methylocaldum'|Genus=
='g__Methylocapsa'|Genus=='g__Methylocella'|Genus=='g__Methylococcus'|Genus=='g__Methylocystis'
|Genus=='g__Methyloferula'|Genus=='g__Methylogaea'|Genus=='g__Methyloglobulus'|Genus=='g__Met
hylohalobius'|Genus=='g__Methylomagnum'|Genus=='g__Methylomarinovum'|Genus=='g__Methylomar
inum'|Genus=='g__Methylobacterium-
Methylorubrum'|Genus=='g__Methylorosula'|Genus=='g__Methylomonas'|Genus=='g__Methyloparacoc
cus'|Genus=='g__Methylosarcina'|Genus=='g__Methylosinus'|Genus=='g__Methylosoma'|Genus=='g__
Methylosphaera'|Genus=='g__Methylothermus'|Genus=='g__Methylomirabilis'|Genus=='g__Methyloce a
nibacter'|Genus=='g__Methylogellaceae'|Genus=='g__pLW-
20'|Genus=='g__Methyloligellaceae'|Genus=='g__Methylomicrobium') #Methanogens & methanotrophs



#Non-rarefied data#
methanogen.nr <- subset_taxa(ps_non_rar,
Class =='c__Methanobacteria'|Class =='c__Methanosarcinia'|Order =='o__Methanomassiliicoccales'|Class
=='c__Methanomicrobia'| Order =='o__Methanococcales'|Order =='o__Methanopyrales'|Order
=='Methanocellales'|Class =='Methanofastidiosa')

methanotroph.nr <- subset_taxa(ps_non_rar,
Genus=='g__Clonothrix'|Genus=='g__Crenothrix'|Genus=='g__Methylacidiphilum'|Genus=='g__Methylaci
dimicrobium'|Genus=='g__Methylobacter'|Genus=='g__Methylocaldum'|Genus=='g__Methylocapsa'|Ge
```

nus=='g__Methylocella'|Genus=='g__Methylococcus'|Genus=='g__Methylocystis'|Genus=='g__Methylofe
rula'|Genus=='g__Methylogaea'|Genus=='g__Methyloglobulus'|Genus=='g__Methylohalobius'|Genus=='
g__Methylomagnum'|Genus=='g__Methylomarinovum'|Genus=='g__Methylomarinum'|Genus=='g__Met
hylobacterium-
Methylorubrum'|Genus=='g__Methylorosula'|Genus=='g__Methylomonas'|Genus=='g__Methyloparacoc
cus'|Genus=='g__Methylosarcina'|Genus=='g__Methylosinus'|Genus=='g__Methylosoma'|Genus=='g__
Methylosphaera'|Genus=='g__Methylothermus'|Genus=='g__Methylomirabilis'|Genus=='g__Methyloce
anibacter'|Genus=='g__Methylogellaceae'|Genus=='g__pLW-
20'|Genus=='g__Methyloligellaceae'|Genus=='g__Methylomicrobium')

methano.nr <- subset_taxa(ps_non_rar, Class =='c__Methanobacteria'|Class
=='c__Methanosarcinia'|Order =='o__Methanomassiliicoccales'|Class =='c__Methanomicrobia'|Order
=='o__Methanococcales'|Order =='o__Methanopyrales'|Order =='Methanocellales'|Class
=='Methanofastidiosa'|Genus=='g__Clonothrix'|Genus=='g__Crenothrix'|Genus=='g__Methylacidiphilum'
|Genus=='g__Methylacidimicrobium'|Genus=='g__Methylobacter'|Genus=='g__Methylocaldum'|Genus=
='g__Methylocapsa'|Genus=='g__Methylocella'|Genus=='g__Methylococcus'|Genus=='g__Methylocystis'
|Genus=='g__Methyloferula'|Genus=='g__Methylogaea'|Genus=='g__Methyloglobulus'|Genus=='g__Met
hylohalobius'|Genus=='g__Methylomagnum'|Genus=='g__Methylomarinovum'|Genus=='g__Methylomar
inum'|Genus=='g__Methylobacterium-
Methylorubrum'|Genus=='g__Methylorosula'|Genus=='g__Methylomonas'|Genus=='g__Methyloparacoc
cus'|Genus=='g__Methylosarcina'|Genus=='g__Methylosinus'|Genus=='g__Methylosoma'|Genus=='g__
Methylosphaera'|Genus=='g__Methylothermus'|Genus=='g__Methylomirabilis'|Genus=='g__Methyloce
anibacter'|Genus=='g__Methylogellaceae'|Genus=='g__pLW-
20'|Genus=='g__Methyloligellaceae'|Genus=='g__Methylomicrobium')

#Create sub-objects for compartments and tissues (methanotrophic/methanogenic communities) #

```
methano_wood<- subset_samples(methano,Tissue =='Wood')
methano_leaf<- subset_samples(methano,Tissue =='Leaf')
methano_bark<- subset_samples(methano,Tissue =='Bark')
methano_sw<- subset_samples(methano,Compartment =='Sapwood')
methano_hw<- subset_samples(methano,Compartment =='Heartwood')
methano_epi<- subset_samples(methano_leaf,Compartment =='Leaf_epiphyte')
methano_endo<- subset_samples(methano_leaf,Compartment =='Leaf_endophyte')

methanotroph_leaf<- subset_samples(methanotroph,Tissue =='Leaf')
methanotroph_bark<- subset_samples(methanotroph,Tissue =='Bark')
methanotroph_wood<- subset_samples(methanotroph,Tissue =='Wood')
methanotroph_sw<- subset_samples(methanotroph,Compartment =='Sapwood')
methanotroph_hw<- subset_samples(methanotroph,Compartment =='Heartwood')
methanotroph_epi<- subset_samples(methanotroph_leaf,Compartment =='Leaf_epiphyte')
methanotroph_endo<- subset_samples(methanotroph_leaf,Compartment =='Leaf_endophyte')
methanogen_wood<- subset_samples(methanogen,Tissue =='Wood')
methanogen_leaf<- subset_samples(methanogen,Tissue =='Leaf')
methanogen_bark<- subset_samples(methanogen,Tissue =='Bark')
methanogen_sw<- subset_samples(methanogen,Compartment =='Sapwood')
methanogen_hw<- subset_samples(methanogen,Compartment =='Heartwood')
methanogen_epi<- subset_samples(methanogen_leaf,Compartment =='Leaf_epiphyte')
```

```
methanogen_endo<- subset_samples(methanogen_leaf,Compartment =='Leaf_endophyte')
methanogen_wood<- subset_samples(methanogen,Tissue =='Wood')
```

#ALPHA DIVERSITY ANALYSIS #

 # Alpha diversity of methanogens-methanotrophs by tissue type#

```
Shannon_troph <- estimate_richness(methanotroph, split=TRUE, measures= "Shannon")
Shannon_trophdf <- data.frame(Shannon_troph)
Shannon_trophdf $Tissue <- sample$Tissue
aov <-aov(Shannon ~ Tissue, Shannon_trophdf)
summary(aov)
posthoc <- TukeyHSD(x=aov, 'Tissue', conf.level=0.95)
posthoc
model <- lm(Shannon_troph ~ Tissue, data = Shannon_trophdf)
model_means <- emmeans(object = model,  specs = "Tissue")
model_means_troph <- cld(object = model_means, adjust = "Tukey", Letters = letters, alpha = 0.05)

Shannon_gen <- estimate_richness(methanogen, split=TRUE, measures= "Shannon")
Shannon_gendf <- data.frame(Shannon_gen)
Shannon_gendf $Tissue <- sample$Tissue
aov <-aov(Shannon ~ Tissue, Shannon_gendf)
summary(aov)
posthoc <- TukeyHSD(x=aov, 'Tissue', conf.level=0.95)
posthoc
model <- lm(Shannon_gen ~ Tissue, data = Shannon_gendf)
model_means <- emmeans(object = model,  specs = "Tissue")
model_means_gen <- cld(object = model_means, adjust = "Tukey", Letters = letters, alpha = 0.05)
```

#Plot alpha diversity of methanogens and methanotrophs

```
p1 <- plot_richness(methanogen, x="Tissue", measures = c("Shannon"), color = "Tissue")+
theme_bw()+ geom_boxplot(fill=c("orangered","green4","blue3"))+
scale_colour_manual(values=c("orangered4","darkgreen","midnightblue"))+theme(axis.title =
element_text(size=11, colour = "black"), axis.text.y = element_text(size = 11, colour = "black"), axis.text.x =
element_text(size = 11, colour="black"),legend.text=element_text(size = 11, colour = "black"),
legend.title=element_text(size = 11, colour = "black")) + geom_text(data = model_means_gen, aes(y =
emmean, x = Tissue, label = str_trim(.group)),  hjust = 1.5,vjust =-9,color = "black")

p2 <- plot_richness(methanotroph, x="Tissue", measures = c("Shannon"), color = "Tissue")+
theme_bw()+ geom_boxplot(fill=c("orangered","green4","blue3"))+
scale_colour_manual(values=c("orangered4","darkgreen","midnightblue"))+theme(axis.title =
element_text(size=11, colour = "black"), axis.text.y = element_text(size = 11, colour = "black"), axis.text.x =
element_text(size = 11, colour="black"),legend.text=element_text(size = 11, colour = "black"),
legend.title=element_text(size = 11, colour = "black")) + geom_text(data = model_means_troph, aes(y =
emmean, x = Tissue, label = str_trim(.group)),  hjust = 1.5,vjust =-9,color = "black")

ggarrange(p1, p2, font.label = list(size = 11),labels = c("a","b"),  nrow = 1,col = 2,common.legend =
TRUE, align="hv")
```

#Alpha diversity of methanogens-methanotrophs by species (compartment level)#

```
shannon <- estimate_richness(methanogen_sw, split=TRUE, measures= "Shannon") #Change
'methanogen_sw' with the other phyloseq objects corresponding to methanogen or methanotroph
communities of each compartment#
   shannondf <- data.frame(shannon)
   sample_sw <- sample_data(methanogen_sw)
   shannondf$TreeTreeSpecies <- sample_sw$TreeTreeSpecies
   aov <-aov(Shannon ~ TreeSpecies, shannondf)
   summary(aov)
   posthoc <- TukeyHSD(x=aov, 'TreeSpecies', conf.level=0.95)
   posthoc

   model <- lm(Shannon ~ TreeSpecies, data = shannondf)
   model_means <- emmeans(object = model, specs = "TreeSpecies")
   model_means_cld <- cld(object = model_means, = "Tukey", Letters = letters, alpha = 0.05)
```

#Plot alpha diversity of methanogens and methanotrophs

```
plot_richness(methanogen_sw, x="TreeSpecies", measures = c("Shannon"), color = "TreeSpecies")+
theme_bw()+ geom_boxplot(fill=c("coral2", "darkgreen", "deepskyblue3","violetred2", "red3"))+
scale_colour_manual(values=c("coral3", "forestgreen", "deepskyblue4","violetred3",
"red4"))+theme(axis.text.y = element_text(size = 11, colour = "black"),legend.text=element_text(size = 11,
colour = "black"), legend.title=element_text(size = 11, colour = "black"), axis.text.x=element_blank(),
axis.title.x= element_blank(),axis.ticks.x=element_blank(),axis.title.y= element_blank()) + geom_text(data
= model_means_cld, aes(y = emmean, x = TreeSpecies, label = str_trim(.group)),  hjust = 0.5,vjust =-
0.5,color = "black")
```

#BETA DIVERSITY ANALYSES #

```
    ps_bray <- phyloseq::distance(methano, method = "bray")
    sampledf <- data.frame(sample_data(methano))
    sampledf
    set.seed(123)
 permanova1 <- adonis2(formula= ps_bray ~  FloodFrequency + TreeSpecies + Tissue + Compartment +
TreeSpecies*FloodFrequency + TreeSpecies*Compartment+Block/FloodFrequency + Tissue/Compartment,
strata= sampledf$Block, data = sampledf, by = "terms") #strata= sampledf$Block
   permanova1
permanova2 <- adonis2(formula= ps_bray ~  FloodFrequency + TreeSpecies + Tissue + Compartment
                  + TreeSpecies*FloodFrequency + TreeSpecies*Tissue
                  +Block/FloodFrequency + Tissue/Compartment,strata= sampledf$Block, data = sampledf, by
= "terms") #strata= sampledf$Block
   permanova2
   AICc_permanova2(permanova1)
   AICc_permanova2(permanova2)
```

#NMDS ORDINATION #

```
    ord <- ordinate (methano, "NMDS", "bray") #Change to ps for 16S community

    ord.sampling.point_plot <- plot_ordination(methano, ord, type = "samples", color ="Tissue",
shape="Compartment")
```

```
ord.sampling.point_plot + geom_point(size = 2) + stat_ellipse(aes(group= Tissue)) +
scale_colour_manual(values = c("orangered","green4", "blue3")) +
 scale_shape_manual(values = c(16, 18, 17,15,8))+ theme_bw() + theme(axis.title =
element_text(size=12, colour = "black"), axis.text.y = element_text(size = 12, colour = "black"), axis.text.x =
element_text(size = 12, colour="black"),legend.text=element_text(size = 12, colour = "black"),
legend.title=element_text(size = 12, colour = "black"))

ord.sampling.point_plot <- plot_ordination(ps, ord, type = "samples", color ="TreeSpecies",
shape="Tissue")
ord.sampling.point_plot + geom_point(size = 2) + scale_colour_manual(values = c("coral2",
"forestgreen", "deepskyblue3","violetred2", "red3")) +
 scale_shape_manual(values = c(16, 17,15))+ theme_bw() + theme(axis.title = element_text(size=12,
colour = "black"), axis.text.y = element_text(size = 12, colour = "black"), axis.text.x = element_text(size =
12, colour="black"),legend.text=element_text(size = 12, colour = "black"), legend.title=element_text(size =
12, colour = "black"))
```

 # PAIRWISE-ADONIS #

#Test which species differ in their methanotrophic/methanogenic communities for each compartment #

```
 ps_bray <- phyloseq::distance(methano_hw, method = “bray”)
sampledf <- data.frame(sample_data(methano_hw)) #Change “methano_hw’ with the other phyloseq
objects corresponding to methanogen-methanotroph communities of each compartment#
permanova <- adonis2(ps_bray ~ TreeSpecies, data = sampledf)
permanova
pairwise.adonis(ps_bray, sample_data(methano_hw)$TreeSpecies,p.adjust.m ='bonferroni')
```

#DIFFERENTIAL ABUNDANCE ANALYSIS: ANCOMBC

 # Differential abundance tests for methanogen/methanotroph genera between tree tissues (To do on the phyloseq objects created from the non-rarefied data)

```
 samplenr$Tissue = factor(samplenr$Tissue, levels = c("Bark","Leaf", "Wood")) #Change the 1st group
to test differential abundance according to different reference group
 sample_data(methano.nr) <- samplenr

 out = ancombc2(group = "Tissue", data = methano.nr,tax_level= "Genus", fix_formula= "Tissue",
p_adj_method = "holm",struc_zero = FALSE, neg_lb = TRUE)
 res = out$res
```

 # Differential abundance tests for methanogen/methanotroph genera of each tissue between tree species (To do on the phyloseq objects created from the non-rarefied data)

```
 meta_wood <- read.csv("meta_wood.csv")# Load the metadata file of the tissue
 meta_wood <- data.frame(meta_wood, row.names = "SampleID")
 sample_hw<- subset(meta_wood, meta_wood$Compartment =='Heartwood')
 sample_hw$TreeSpecies = factor(sample_hw$TreeSpecies, levels = c("Salix","Fraxinus", "Populus",
"Ulmus","Acer")) #Change the 1st species to test differential abundance according to different species as
reference
```

```
        sample_data(methano_hw.nr) <- sample_hw
        sample_sw<- subset(meta_wood, meta_wood$Compartment =='Sapwood')
        sample_sw$TreeSpecies = factor(sample_sw$TreeSpecies, levels = c("Salix","Fraxinus", "Populus",
"Ulmus","Acer"))
        sample_data(methano_sw.nr) <- sample_sw

        meta_leaf <- read.csv("meta_leaf.csv")
        meta_leaf <- data.frame(meta_leaf, row.names = "SampleID")
        meta_endo <- subset(meta_leaf, meta_leaf$Compartment =='LEAF_endophyte')
        meta_endo$TreeSpecies = factor(meta_endo$TreeSpecies, levels = c("Fraxinus","Ulmus", "Salix",
"Acer", "Populus"))
        sample_data(methano_endo.nr) <- meta_endo

        meta_epi <- subset(meta_leaf, meta_leaf$Compartment =='LEAF_epiphyte')
        meta_epi$TreeSpecies = factor(meta_epi$TreeSpecies, levels =
c("Fraxinus","Acer","Populus","Ulmus", "Salix"))
        sample_data(methano_epi.nr) <- meta_epi

        meta_bark <- read.csv("meta_bark.csv")
        meta_bark <- data.frame(meta_bark, row.names = "SampleID")
        meta_bark$TreeSpecies = factor(meta_bark$TreeSpecies, levels = c("Salix","Acer","Fraxinus",
"Ulmus","Populus"))
        sample_data(methano_bark.nr) <- meta_bark
out = ancombc2(group = "TreeSpecies", data = methano_bark.nr,tax_level= "Genus", fix_formula=
"TreeSpecies", p_adj_method = "holm",struc_zero = FALSE, neg_lb = TRUE, prv_cut=0.01) #To do on the
object created from the non-rarefied data. Change 'methano_bark.nr' with the other phyloseq objects
representing methanogen-methanotroph communities of the different compartments #
         res = out$res
```

 # BARPLOTS of methanogens & methanotrophs communities #

```
    bark = merge_samples(bark, "TreeSpecies") #Merge phyloseq samples at tree species. Change ‘bark’
with the other phyloseq objects representing microbial communities of the other compartments#
    meta_merged <- read.csv("meta_all_merged.csv")
    meta_merged <- data.frame(meta_merged, row.names = "Species")
    sample_data(bark) <- meta_merged
    x <- bark
    x <- transform_sample_counts(x, function(OTU) OTU/sum(OTU))
    ps.taxglomGENRE <- tax_glom(x, taxrank = "Genus") #merge taxonomic data at the Genus level
    y <- psmelt(ps.taxglomGENRE)
```

**   # Generate relative abundance tables for methanogen and methanotroph genera**

```
    bk_table_methanogens <- subset(y, Class =='c__Methanobacteria'|Class
=='c__Methanosarcinia'|Order =='o__Methanomassiliicoccales'|Class =='c__Methanosarcinia'|
                            Class =='c__Methanomicrobia')
```

```
    bk_table_methanotroph <- subset(y, Order
==’o__Methylococcales’|Genus==’g__Methyloceanibacter’|Genus
==’g__Methylocystis’|Genus==’g__Methyloligellaceae’|
                        Genus ==’g__Methylocella’|Genus==’g__Methylomonas’|Genus
==’g__Methylobacterium-
Methylorubrum’|Genus==’g__Methylorosula’|Genus==’g__Methylobacter’|Genus==’g__Methylomicrobiu
m’|Genus==’g__Methylocapsa’|
                        Genus==’g_Methylovirgula’|Genus==’g_Albibacter’|Genus==’g_Rokubacteriales’)

    y4<-bk_table_methanogens
    y4$Genus <- as.character(y4$Genus)
    y4$Genus[y4$Genus==’g__uncultured’] <- "f__Methanomethylophilaceae"
    colourCount = length(unique(y4$Genus))

    ggplot(data=y4, aes(x=TreeSpecies, y=Abundance, fill=Genus))+ geom_bar(aes(), stat="identity",
position= position_stack(reverse = T)) + theme_bw() +
      theme(axis.text.x = element_text(angle = 90, size=11, colour="black"),axis.text.y =
element_text(size=11, colour="black"), axis.title=element_text(size=11, colour="black"),
legend.text=element_text(size = 11, colour = "black")) +
      scale_fill_manual(values
=c("midnightblue","blue3","deepskyblue4","deepskyblue3","palegreen3","palegreen4", "forestgreen",
"darkgreen","grey33"))+
      labs(y="RA", x="")+ scale_y_continuous(labels = scales::number_format(accuracy = 0.001,
decimal.mark = '.'))

    y5<-bk_table_methanotroph
    y5$Genus <- as.character(y5$Genus)
    colourCount = length(unique(y5$Genus))

    ggplot(data=y5, aes(x=TreeSpecies, y=Abundance, fill=Genus))+ geom_bar(aes(), stat="identity",
position= position_stack(reverse = T)) + theme_bw() +
      theme(axis.text.x = element_text(angle = 90, size=11, colour="black"),axis.text.y =
element_text(size=11, colour="black"), axis.title=element_text(size=11, colour="black"),
legend.text=element_text(size = 11, colour = "black")) +
      scale_fill_manual(values =c("Salmon3", "coral2", "orange2","yellow3","yellowgreen", "chartreuse3",
"springgreen4","deepskyblue3", "deepskyblue2", "royalblue",
"slateblue","purple2","purple4","deeppink3","red4", "darkred"))+
      labs(y="RA", x="")+ scale_y_continuous(labels = scales::number_format(accuracy = 0.001,
decimal.mark = '.'))
```

 # Correlations between methanotrophs/methanogens and tree traits #

#MANTEL test for correlations between methanotroph/methanogen community composition and traits #

```
sampledf <- data.frame(sample_data(methano_bark))
env= sampledf[,c(8:10)] #Select columns corresponding to traits to test
ps_bray <- phyloseq::distance(methano_bark, method = "bray")
scale.env = scale(env, center = TRUE, scale = TRUE)
dist.env = dist(scale.env, method = "euclidean")
mantel(ps_bray, dist.env, method = "spearman", permutations = 9999, na.rm = TRUE)
```

#Change 'methano_bark' with the other phyloseq objects corresponding to methanogen-methanotroph communities of each compartment#

 # Linear regression models of methanogens & methanotrophs RA according to traits

 # Stepwise regression to find the best model

```
RA_traits <- read.csv("RA_traits.csv")
model_data <- [,c("BK_methanotrophsRA", "Block", "FloodFrequency", "TreeSpecies", "BK_pH", " DBH ", "BK_H", "BK_D")] #Change the variables for models of the other tissues#
    model_data <- na.omit(model_data)
    full.model <- lm(BK_methanotrophsRA ~ Block + FloodFrequency + TreeSpecies + BK_pH + DBH + BK_H + BK_D, data = model_data, na.action= na.exclude)
    step.model <- stepAIC(full.model, direction = "both")
    summary(step.model)
    shapiro.test(step.model[['residuals']])
```

#LF epi-methanogens RA

```
    epi_gens<- lm(formula = epi_methanogens_RA ~ Block + TreeSpecies + DBH + LMA, data = RA_traits, na.action = na.exclude)
    summary(epi_gens)
    AICc(epi_gens)
    shapiro.test(epi_gens[['residuals']])
    calc.relimp(epi_methanogens_RA ~ factor(Species) +  DHP + MLA, data = RA_traits, type = "lmg")
    emm = emmeans(epi_gens, specs = pairwise ~ Species)
    emm
```

#LF epi-methanotrophs RA

```
    epi_trophe<- lm(formula = epi_methanotrophsRA ~ LF_pH, data = RA_traits, na.action = na.exclude)
    summary(epi_trophe)
    shapiro.test(epi_trophe[['residuals']])
    AICc(epi_trophe)
    calc.relimp(endo_methanotrophsRA ~ factor(Block) +  DBH + LF_H + MLA, data = RA_traits, type = "lmg")
```

#LF endo-methanotrophs RA

```
    endo_troph<- lm(formula = endo_methanotrophsRA ~ Block + TreeSpecies + DBH + LF_H +
                MLA, data = RA_traits, na.action = na.exclude)
```

```
    summary(endo_troph)
    shapiro.test(endo_troph[['residuals']])
    AICc(endo_trophe)
    calc.relimp(endo_methanotrophsRA ~ factor(Block) +  DBH + LF_H + MLA, data = RA_traits, type =
"lmg")
    emm = emmeans(endo_trophe, specs = pairwise ~ TreeSpecies)
    emm

    #SW methanotrophs RA
    sw_trophe = lm(formula = SW_methanotrophsRA ~ SW_H, data = RA_traits, na.action = na.exclude)
    summary(sw_trophe)
    AICc(sw_trophe)
    RA_traits_sw_trophe <- RA_traits[ !is.na(RA_traits$SW_methanotrophsRA), ]

    #SW methanogens RA
    sw_gens = lm(formula = SW_methanogensRA ~ TreeSpecies + DBH + SW_H, data = RA_traits, na.action
= na.exclude)
    summary(sw_gens)
    AICc(sw_gens)
    calc.relimp(HW_methanogensRA ~ factor(TreeSpecies) + SW_H + DBH, data = RA_traits, type = "lmg")
    emm = emmeans(sw_gens, specs = pairwise ~ TreeSpecies)
    emm

    #HW methanogens RA
    hw_gens = lm(formula = HW_methanogensRA ~ FloodFrequency + HW_pH + DBH + HW_H, data =
RA_traits, na.action = na.exclude)
    summary(hw_gens)
    AICc(hw_gens)
    calc.relimp(HW_methanogensRA ~ factor(FloodFrequency) + HW_H + HW_pH+ DBH, data = RA_traits,
type = "lmg")

    #HW methanotrophs RA
    hw_trophs<- lm(formula = HW_methanotrophsRA ~ Block + TreeSpecies + HW_pH + DBH +
            HW_H, data = RA_traits, na.action = na.exclude)
    summary(hw_trophs)
    AICc(hw_trophs)
    calc.relimp(HW_methanotrophsRA ~ factor(TreeSpecies) + HW_H + HW_pH+ DBH, data = RA_traits,
type = "lmg")
    emm = emmeans(hw_trophs, specs = pairwise ~ Block)#yama > pier p=0.06
    emm

    #BK methanotrophs RA
    bk_trophe <- lm(formula = BK_methanotrophsRA ~ TreeSpecies + BK_pH, data = RA_traits, na.action =
na.exclude)
    summary(bk_trophe)
    shapiro.test(bk_trophe[['residuals']])
    calc.relimp(BK_methanotrophsRA ~ factor(TreeSpecies) + BK_pH, data = RA_traits, type = "lmg")
    emm = emmeans(bk_trophe, specs = pairwise ~ TreeSpecies)
    emm
```

```
AICc(bk_trophe)
```

#Plot the correlations between methanogens-methanotrophs RA and traits in models

```
library(sjPlot)
library(sjlabelled)
library(sjmisc)
forest_plot_LF <- plot_models(epi_gens,epi_trophe,endo_trophe,
                  m.labels= c("Epiphyte methanogens", "Epiphyte methanotrophs", "Endophyte
methanotrophs"), legend.title = "Models",
                  axis.labels = c("Humidity", "pH", "LMA","DBH"),
                  show.values = FALSE, show.p = FALSE, p.shape = TRUE,
                  std.est="std", transform = NULL,
                  colors = c("red","darkorange1","green4"),
                  spacing = 0.6, vline.color="black",
                  title = "" , rm.terms = c("Site [Dupa,Maskinonge,Pierreville,Yamachiche,
Nicolet]","Species [Fraxinus,Populus,Salix,Ulmus]")) + ylim(-1, 1) + theme_minimal() +
  theme(legend.text=element_text(size=11), axis.title.y=element_text(size=11,colour="black"),
axis.title.x=element_text(size=11,colour="black"),
axis.text.x=element_text(size=11,colour="black"),axis.text.y=element_text(size=11, colour="black"))

forest_plot_wood <- plot_models(sw_gens,sw_trophe,hw_gens,hw_trophs,bk_trophe,
                   m.labels= c("SW methanogens", "SW methanotrophs","HW methanogens", "HW
methanotrophs","BK methanotrophs"),
                   legend.title = "Models",
                   axis.labels = c("BK pH","HW Humidity", "HW pH", "SW Humidity", "DBH"),
                   show.values = FALSE, show.p = FALSE, p.shape = TRUE,
                   std.est="std", transform = NULL, colors =
c("red3","darkorange2","darkgreen","darkgoldenrod2","chartreuse3"),
                   spacing = 0.6, vline.color="black",
                   title = "" , rm.terms = c("Site [Dupa,Maskinonge,Pierreville,Yamachiche,
Nicolet]","Species [Fraxinus,Populus,Salix,Ulmus]","FloodFrequency [LFF]")) +
  ylim(-1, 1) +
  theme_minimal() +
  theme(legend.text=element_text(size=11), axis.title.y=element_text(size=11,colour="black"),
axis.title.x=element_text(size=11,colour="black"),
axis.text.x=element_text(size=11,colour="black"),axis.text.y=element_text(size=11, colour="black"))

Forestplot <- ggarrange(forest_plot_LF,forest_plot_wood, font.label = list(size = 11),labels = c("a","b"),
nrow = 2,  ncol = 1,common.legend = F,align="hv")
```