Soil microbial biodiversity: Bacteria abundant taxa

Mercedes Roman

2022-07-16

## Soil bacteria - Abundant taxa

We transformed the abundance data into relative abundance and calculated the average relative abundance of each OTU across all sites. The abundant taxa were defined as those from the top 1%, after ranking the OTUs by relative abundance, that occur in at least 10 % sites of sites or those OTUs that occurred in 50% of sites. The dominant taxa accounted mostly between 10-30 % of the OTUs present in the samples.Theese criteria resulted in 2063 taxa accross 1373 samples.

abun.ps

## phyloseq-class experiment-level object  
## otu\_table() OTU Table: [ 2063 taxa and 1373 samples ]:  
## sample\_data() Sample Data: [ 1373 samples by 117 sample variables ]:  
## tax\_table() Taxonomy Table: [ 2063 taxa by 7 taxonomic ranks ]:  
## taxa are rows

We will test three methods for analyzing the biodiversity: non-metric multidimensional scaling (NMDS), Copula ordination, and Umap.

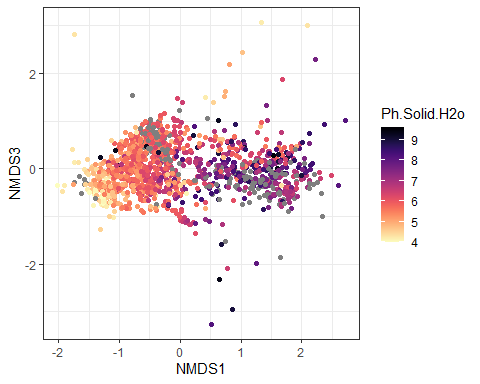
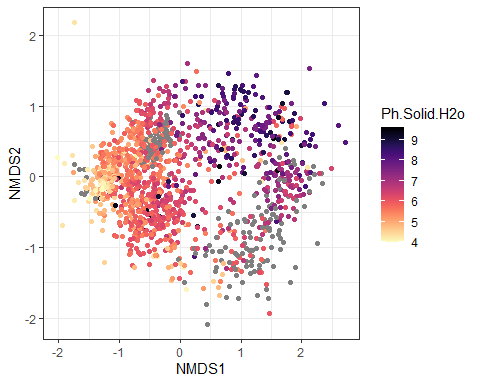
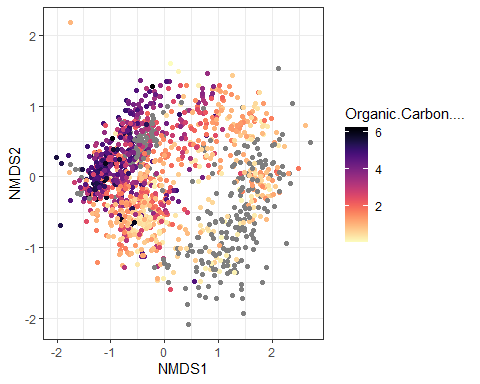
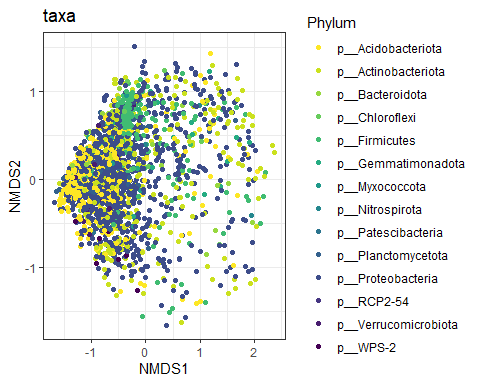
### NMDS

First I perform a NMDS using Bray-Curtis dissimilarity metric on abundant taxa.

set.seed(1812)  
Bacteria.Abun.NMDS = ordinate(abun.ps, "NMDS", "bray",   
 ### Bray-Curtis dissimilarity  
 k=3,   
 ### 3 dimensions  
 try=20, trymax=2000,   
 ### Min 20 random starts, max 2000  
 maxit=5000,   
 ### Try 5000 iterations at each start  
 smin = 0.05,   
 ### I would accept a stress of 0.1, the structure of the community is well represented so I set the threshold in 0.05.  
 trace = FALSE, plot=FALSE)  
Bacteria.Abun.NMDS

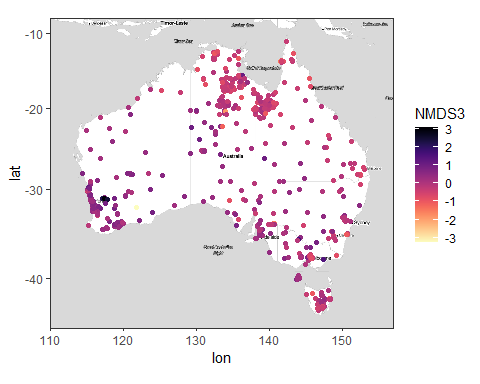
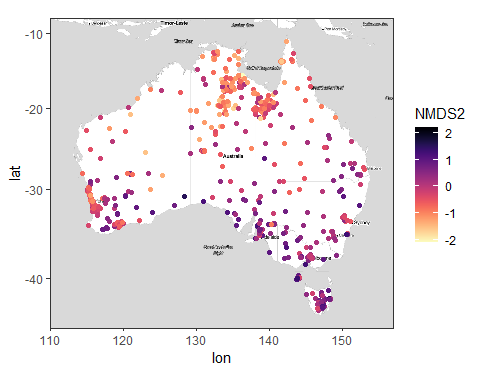
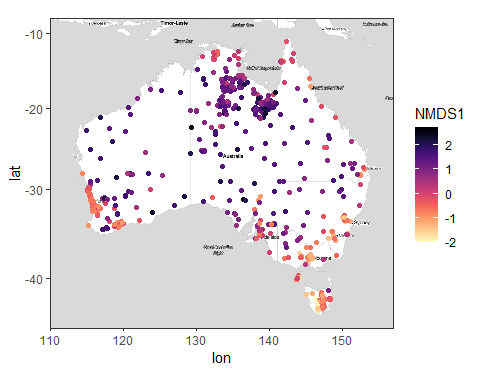
##   
## Call:  
## metaMDS(comm = veganifyOTU(physeq), distance = distance, k = 3, try = 20, trymax = 2000, trace = FALSE, plot = FALSE, maxit = 5000, smin = 0.05)   
##   
## global Multidimensional Scaling using monoMDS  
##   
## Data: veganifyOTU(physeq)   
## Distance: bray   
##   
## Dimensions: 3   
## Stress: 0.1162528   
## Stress type 1, weak ties  
## No convergent solutions - best solution after 2000 tries  
## Scaling: centring, PC rotation, halfchange scaling   
## Species: expanded scores based on 'veganifyOTU(physeq)'

save(Bacteria.Abun.NMDS,file=paste0(OutDir,"Bacteria.Abun.NMDS.RData"))



The stress of the NMDS is 0.11, suggesting that the structure of the data has been captured relatively well.

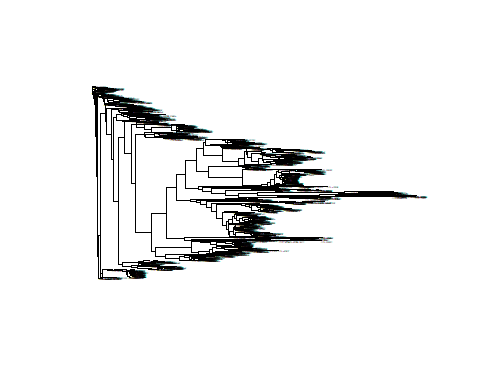
There are some spatial patterns, which would be interesting to know if they respond to environmental facotrs, latitude gradient, or dataset of origin (BASE or GA).



#### NMDS with weighted unifrac distance

Instead of running the NMDS with weighted unifrac distance directly in physoleq, I first calculate the weighted unifrac distance with phyloseq and the NMDS with the vegan package so I can set my preferred parameters. This NMDS analysis does find a convergent solution with a stress of 0.09, which indicates that the structure of the data has been captured well.

### Plot the tree  
plot(fitGTR$tree, cex=0.1)



cl <- makeCluster(6)  
registerDoParallel(cl)  
set.seed(2665)  
uf\_BacAbun <- UniFrac(Bac\_tree, weighted=TRUE, normalized=TRUE, parallel=TRUE, fast=TRUE)  
stopCluster(cl)  
  
class(uf\_BacAbun)

## [1] "dist"

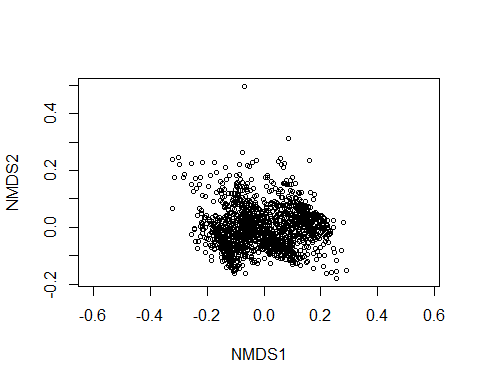
dim(uf\_BacAbun)

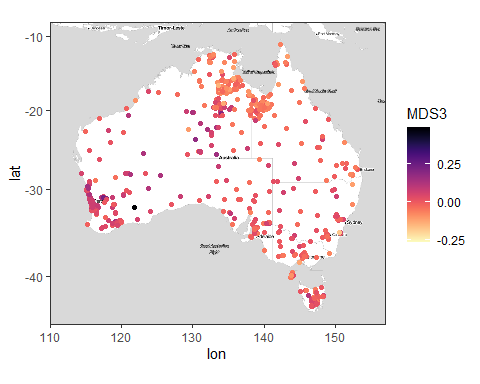
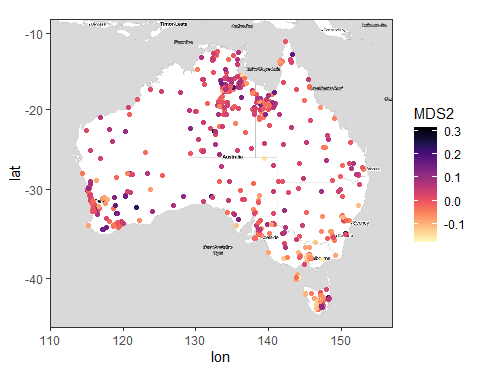
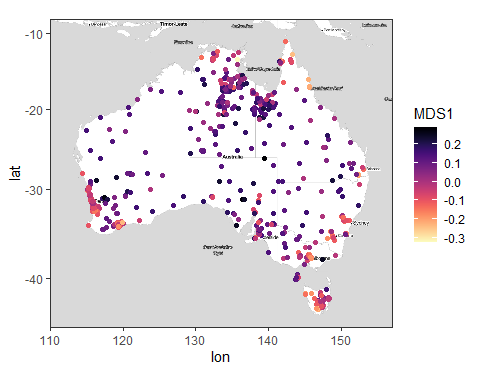
## [1] 1373 1373

set.seed(5436)  
nmds.wuf <- vegan::metaMDS(comm = uf\_BacAbun, k = 3,   
 try = 50, trymax = 2000,  
 trace = FALSE,   
 plot = FALSE,   
 maxit=2000,  
 smin = 0.05)  
  
nmds.wuf

##   
## Call:  
## vegan::metaMDS(comm = uf\_BacAbun, k = 3, try = 50, trymax = 2000, trace = FALSE, plot = FALSE, maxit = 2000, smin = 0.05)   
##   
## global Multidimensional Scaling using monoMDS  
##   
## Data: uf\_BacAbun   
## Distance: user supplied   
##   
## Dimensions: 3   
## Stress: 0.08999892   
## Stress type 1, weak ties  
## Two convergent solutions found after 146 tries  
## Scaling: centring, PC rotation   
## Species: scores missing

ordiplot(nmds.wuf)





## Copulas models

We start by bringing the total counts

ps\_srf.r

## phyloseq-class experiment-level object  
## otu\_table() OTU Table: [ 568189 taxa and 1373 samples ]:  
## sample\_data() Sample Data: [ 1373 samples by 117 sample variables ]:  
## tax\_table() Taxonomy Table: [ 568189 taxa by 7 taxonomic ranks ]:  
## taxa are rows

abunA.ps <- prune\_taxa(taxa\_names(ps\_srf.r)%in% taxa\_names(Bac\_tree),ps\_srf.r)  
abunA.ps <- prune\_samples(sample\_sums(abunA.ps)>0, abunA.ps)  
abunA.ps

## phyloseq-class experiment-level object  
## otu\_table() OTU Table: [ 2063 taxa and 1373 samples ]:  
## sample\_data() Sample Data: [ 1373 samples by 117 sample variables ]:  
## tax\_table() Taxonomy Table: [ 2063 taxa by 7 taxonomic ranks ]:  
## taxa are rows

#Prepare the data in the format for the copulas models  
ta <- abunA.ps %>% otu\_table %>% as.matrix() %>% t() %>% as.data.frame()  
ta[1:10,1:10]

## OTU.34 OTU.44 OTU.47 OTU.340 OTU.370 OTU.668 OTU.964  
## SampleID.102.100.100/12620 0 1 10 0 0 10 0  
## SampleID.102.100.100/13896 0 0 2 0 0 4 0  
## SampleID.102.100.100/8150 0 0 0 0 8 69 0  
## SampleID.102.100.100/19487 8 0 3 2 0 17 0  
## SampleID.102.100.100/42146 4 0 5 0 1 60 0  
## SampleID.102.100.100/15981 3 0 3 0 1 148 0  
## SampleID.102.100.100/39232 0 0 0 0 0 2 0  
## SampleID.102.100.100/12568 0 0 0 0 0 23 0  
## SampleID.102.100.100/42248 5 0 3 0 0 19 0  
## SampleID.102.100.100/42290 0 0 1 0 0 7 0  
## OTU.1371 OTU.1384 OTU.1901  
## SampleID.102.100.100/12620 0 2 0  
## SampleID.102.100.100/13896 0 9 0  
## SampleID.102.100.100/8150 0 1 0  
## SampleID.102.100.100/19487 6 0 0  
## SampleID.102.100.100/42146 0 0 0  
## SampleID.102.100.100/15981 0 0 0  
## SampleID.102.100.100/39232 0 0 0  
## SampleID.102.100.100/12568 0 0 0  
## SampleID.102.100.100/42248 0 1 3  
## SampleID.102.100.100/42290 0 0 4

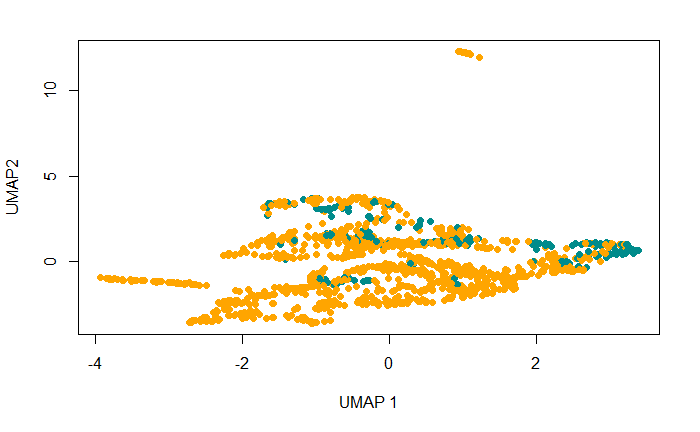
context.data <- data.frame(sample\_data(abunA.ps))  
context.data$Dataset <- ifelse(is.na(context.data$Ph.Solid.H2o), "GA", "BASE")  
rownames(context.data) <- 1:nrow(context.data)  
rownames(ta) <- 1:nrow(ta)

We can check the optimal number of latent factors, up to a number that we consider may be useful to map (maximum 5). We check the BIC of the ordination for 2 to 5 latent factors.

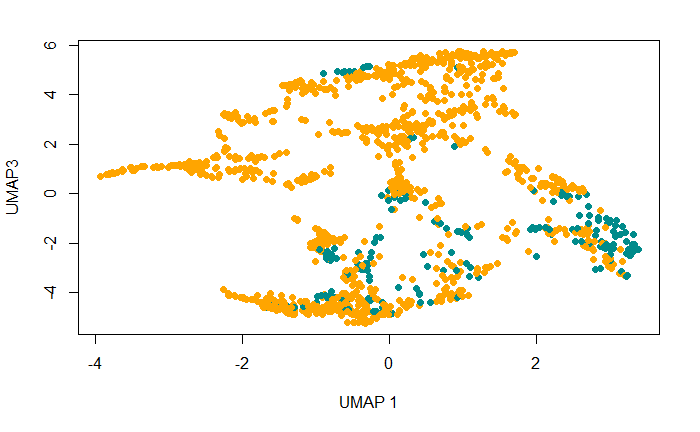
NOTE: I cannot run the copulas models for bacteria in my desktop. Waiting to have installed some R packages in Artemis so I can run them there with enough memory

## UMAP

library(umap)  
  
### Tune parameters  
custom.config = umap.defaults  
custom.config$n\_components=3  
custom.config$random\_state = 1984  
bacteria.umap = umap(as.matrix(uf\_BacAbun),config=custom.config, input="dist")  
  
### Plot  
umap.context <- sample\_data(Bac\_tree)  
umap.context$Dataset <- ifelse(is.na(umap.context$Ph.Solid.H2o), "GA", "BASE")  
umap.context$Dataset <- as.factor(umap.context$Dataset)  
  
### Attach the umap scores  
umap.context$Umap1 <- bacteria.umap$layout[,1]  
umap.context$Umap2 <- bacteria.umap$layout[,2]  
umap.context$Umap3 <- bacteria.umap$layout[,3]  
#with(nmds.wuf.context, levels(Dataset))  
  
data.origin <- ifelse(umap.context$Dataset=="GA", "darkcyan", "orange")   
  
plot(x=umap.context$Umap1, y=umap.context$Umap2 , col=data.origin, pch=19, xlab="UMAP 1", ylab="UMAP2")

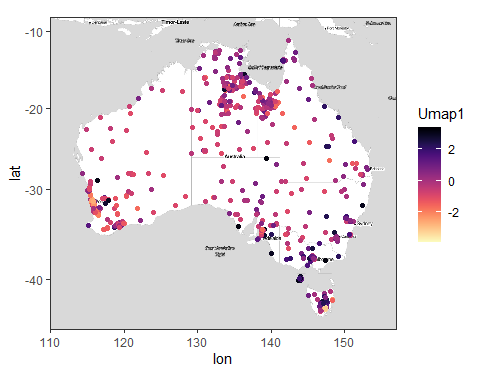


plot(x=umap.context$Umap1, y=umap.context$Umap3 , col=data.origin, pch=19, xlab="UMAP 1", ylab="UMAP3")

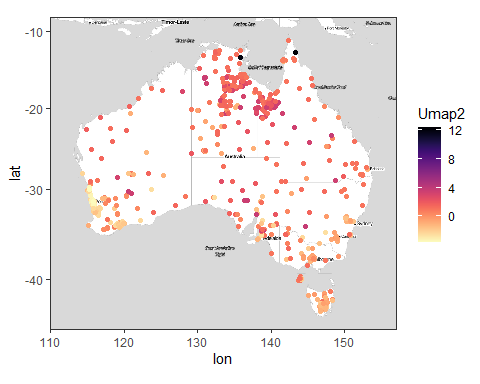


save(bacteria.umap,umap.context, file=paste0(OutDir,"Bacteria.umap.wuf.RData"))

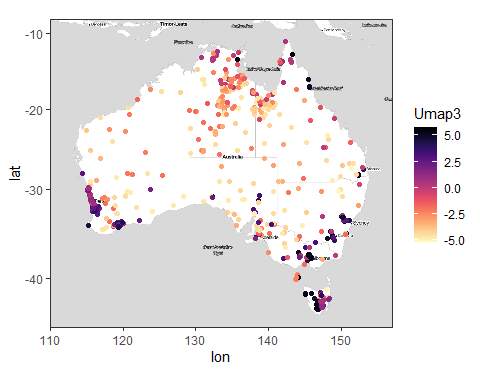
ggmap(AusMap) + geom\_point(aes(y = Latitude, x =Longitude,color=Umap1), data = umap.context) +  
 scale\_color\_viridis(discrete = FALSE, option="A", direction = -1)



ggmap(AusMap) + geom\_point(aes(y = Latitude, x =Longitude,color=Umap2), data = umap.context) +  
 scale\_color\_viridis(discrete = FALSE, option="A", direction = -1)



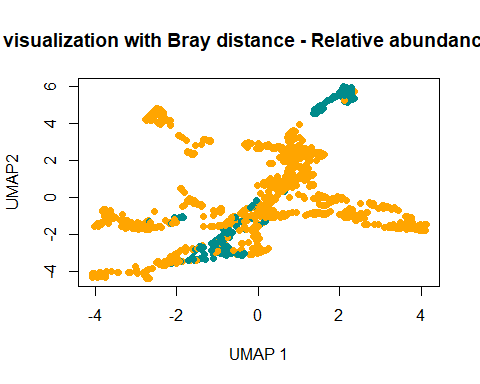
ggmap(AusMap) + geom\_point(aes(y = Latitude, x =Longitude,color=Umap3), data = umap.context) +  
 scale\_color\_viridis(discrete = FALSE, option="A", direction = -1)



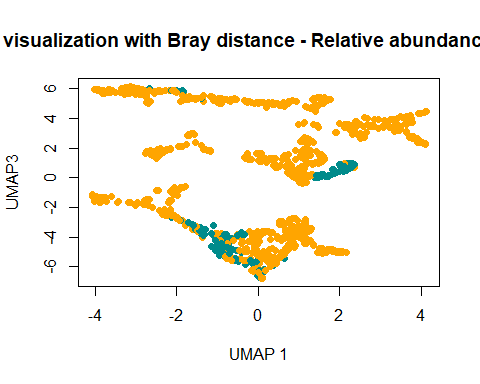
### I calculate first Bray distances  
  
#Prepare the relative abundance in the right format  
ra <- Bac\_tree %>% otu\_table %>% as.matrix() %>% t() %>% as.data.frame()  
ra[1:10,1:10]

## OTU.34 OTU.44 OTU.47 OTU.340  
## SampleID.102.100.100/12620 0.0000000000 9.917683e-05 9.917683e-04 0.0000000000  
## SampleID.102.100.100/13896 0.0000000000 0.000000e+00 1.983537e-04 0.0000000000  
## SampleID.102.100.100/8150 0.0000000000 0.000000e+00 0.000000e+00 0.0000000000  
## SampleID.102.100.100/19487 0.0007934147 0.000000e+00 2.975305e-04 0.0001983537  
## SampleID.102.100.100/42146 0.0003967073 0.000000e+00 4.958842e-04 0.0000000000  
## SampleID.102.100.100/15981 0.0002975305 0.000000e+00 2.975305e-04 0.0000000000  
## SampleID.102.100.100/39232 0.0000000000 0.000000e+00 0.000000e+00 0.0000000000  
## SampleID.102.100.100/12568 0.0000000000 0.000000e+00 0.000000e+00 0.0000000000  
## SampleID.102.100.100/42248 0.0004958842 0.000000e+00 2.975305e-04 0.0000000000  
## SampleID.102.100.100/42290 0.0000000000 0.000000e+00 9.917683e-05 0.0000000000  
## OTU.370 OTU.668 OTU.964 OTU.1371  
## SampleID.102.100.100/12620 0.000000e+00 0.0009917683 0 0.000000000  
## SampleID.102.100.100/13896 0.000000e+00 0.0003967073 0 0.000000000  
## SampleID.102.100.100/8150 7.934147e-04 0.0068432014 0 0.000000000  
## SampleID.102.100.100/19487 0.000000e+00 0.0016860061 0 0.000595061  
## SampleID.102.100.100/42146 9.917683e-05 0.0059506099 0 0.000000000  
## SampleID.102.100.100/15981 9.917683e-05 0.0146781712 0 0.000000000  
## SampleID.102.100.100/39232 0.000000e+00 0.0001983537 0 0.000000000  
## SampleID.102.100.100/12568 0.000000e+00 0.0022810671 0 0.000000000  
## SampleID.102.100.100/42248 0.000000e+00 0.0018843598 0 0.000000000  
## SampleID.102.100.100/42290 0.000000e+00 0.0006942378 0 0.000000000  
## OTU.1384 OTU.1901  
## SampleID.102.100.100/12620 1.983537e-04 0.0000000000  
## SampleID.102.100.100/13896 8.925915e-04 0.0000000000  
## SampleID.102.100.100/8150 9.917683e-05 0.0000000000  
## SampleID.102.100.100/19487 0.000000e+00 0.0000000000  
## SampleID.102.100.100/42146 0.000000e+00 0.0000000000  
## SampleID.102.100.100/15981 0.000000e+00 0.0000000000  
## SampleID.102.100.100/39232 0.000000e+00 0.0000000000  
## SampleID.102.100.100/12568 0.000000e+00 0.0000000000  
## SampleID.102.100.100/42248 9.917683e-05 0.0002975305  
## SampleID.102.100.100/42290 0.000000e+00 0.0003967073

bray.dist <- vegan::vegdist(ra, "bray")  
  
library(umap)  
### Tune parameters  
custom.config = umap.defaults  
custom.config$n\_components=3  
custom.config$random\_state = 1984  
  
### Perform UMAP  
bacteria.umap.bray = umap(as.matrix(bray.dist),config=custom.config, input="dist")  
  
### Plot  
umap.context <- sample\_data(Bac\_tree)  
umap.context$Dataset <- ifelse(is.na(umap.context$Ph.Solid.H2o), "GA", "BASE")  
umap.context$Dataset <- as.factor(umap.context$Dataset)  
  
### Attach the umap scores  
umap.context$Umap1 <- bacteria.umap.bray$layout[,1]  
umap.context$Umap2 <- bacteria.umap.bray$layout[,2]  
umap.context$Umap3 <- bacteria.umap.bray$layout[,3]  
#with(nmds.wuf.context, levels(Dataset))  
  
data.origin <- ifelse(umap.context$Dataset=="GA", "darkcyan", "orange")   
  
plot(x=umap.context$Umap1, y=umap.context$Umap2 , col=data.origin, pch=19, xlab="UMAP 1", ylab="UMAP2",  
 main="UMAP visualization with Bray distance - Relative abundance Bacteria")

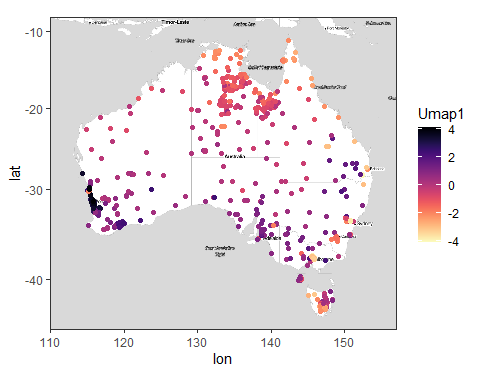


plot(x=umap.context$Umap1, y=umap.context$Umap3 , col=data.origin, pch=19, xlab="UMAP 1", ylab="UMAP3",  
 main="UMAP visualization with Bray distance - Relative abundance Bacteria")

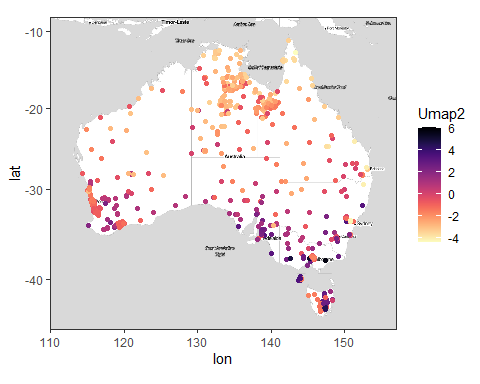


save(bacteria.umap.bray,umap.context, file=paste0(OutDir,"Bacteria.umap.bray.RData"))

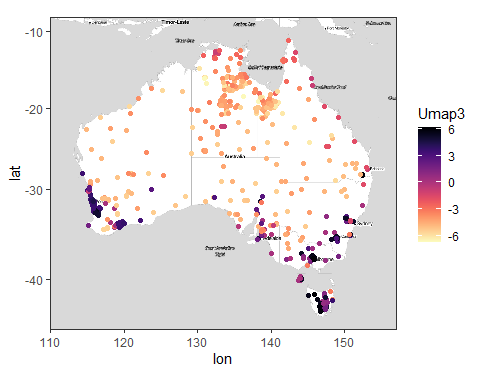
ggmap(AusMap) + geom\_point(aes(y = Latitude, x =Longitude,color=Umap1), data = umap.context) +  
 scale\_color\_viridis(discrete = FALSE, option="A", direction = -1)



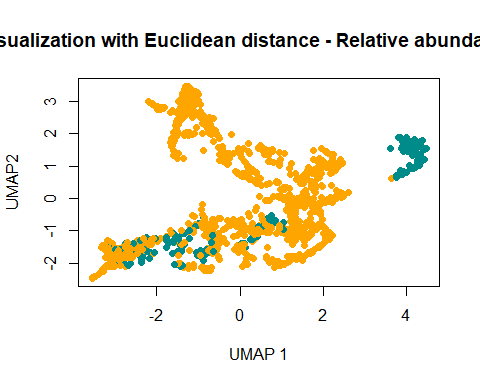
ggmap(AusMap) + geom\_point(aes(y = Latitude, x =Longitude,color=Umap2), data = umap.context) +  
 scale\_color\_viridis(discrete = FALSE, option="A", direction = -1)



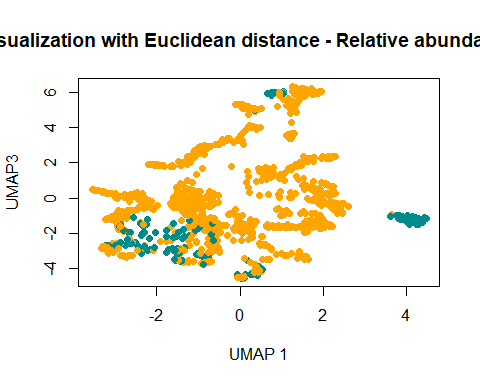
ggmap(AusMap) + geom\_point(aes(y = Latitude, x =Longitude,color=Umap3), data = umap.context) +  
 scale\_color\_viridis(discrete = FALSE, option="A", direction = -1)



### Tune parameters  
custom.config = umap.defaults  
custom.config$n\_components=3  
custom.config$random\_state = 1984  
  
bacteria.umap2 = umap(ra, config=custom.config)  
  
### Plot  
umap.context <- sample\_data(Bac\_tree)  
umap.context$Dataset <- ifelse(is.na(umap.context$Ph.Solid.H2o), "GA", "BASE")  
umap.context$Dataset <- as.factor(umap.context$Dataset)  
  
### Attach the umap scores  
umap.context$Umap1 <- bacteria.umap2$layout[,1]  
umap.context$Umap2 <- bacteria.umap2$layout[,2]  
umap.context$Umap3 <- bacteria.umap2$layout[,3]  
  
data.origin <- ifelse(umap.context$Dataset=="GA", "darkcyan", "orange")   
  
plot(x=umap.context$Umap1, y=umap.context$Umap2 , col=data.origin, pch=19, xlab="UMAP 1", ylab="UMAP2",  
 main="UMAP visualization with Euclidean distance - Relative abundance bacteria")

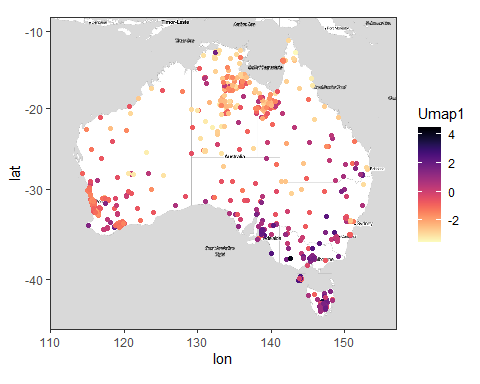


plot(x=umap.context$Umap1, y=umap.context$Umap3 , col=data.origin, pch=19, xlab="UMAP 1", ylab="UMAP3",  
 main="UMAP visualization with Euclidean distance - Relative abundance bacteria")

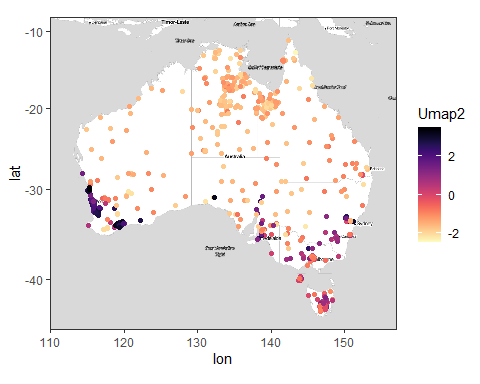


save(bacteria.umap2,umap.context, file=paste0(OutDir,"Bacteria.umap.eucl.RData"))

ggmap(AusMap) + geom\_point(aes(y = Latitude, x =Longitude,color=Umap1), data = umap.context) +  
 scale\_color\_viridis(discrete = FALSE, option="A", direction = -1)



ggmap(AusMap) + geom\_point(aes(y = Latitude, x =Longitude,color=Umap2), data = umap.context) +  
 scale\_color\_viridis(discrete = FALSE, option="A", direction = -1)



ggmap(AusMap) + geom\_point(aes(y = Latitude, x =Longitude,color=Umap3), data = umap.context) +  
 scale\_color\_viridis(discrete = FALSE, option="A", direction = -1)

