Soil microbial biodiversity: Fungi abundant taxa

Mercedes Roman

2022-07-08

## Soil fungi - 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.These criteria resulted in 184 taxa across 1317 samples.

abun.ps

## phyloseq-class experiment-level object  
## otu\_table() OTU Table: [ 184 taxa and 1317 samples ]:  
## sample\_data() Sample Data: [ 1317 samples by 117 sample variables ]:  
## tax\_table() Taxonomy Table: [ 184 taxa by 7 taxonomic ranks ]:  
## taxa are rows

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

## NMDS

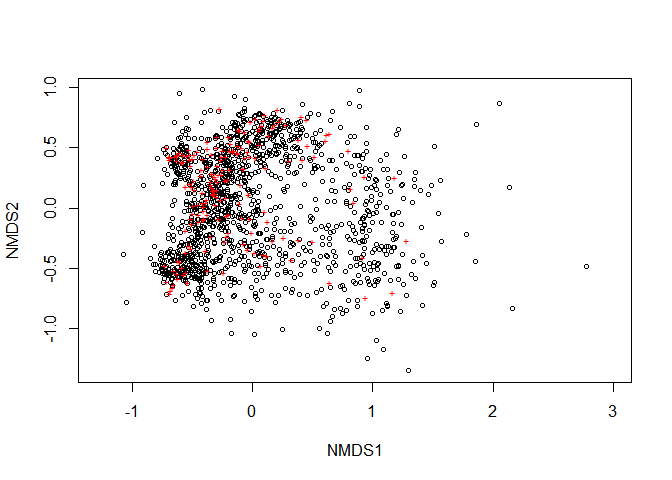
First I perform a NMDS using Bray-Curtis dissimilarity metric on abundant taxa.

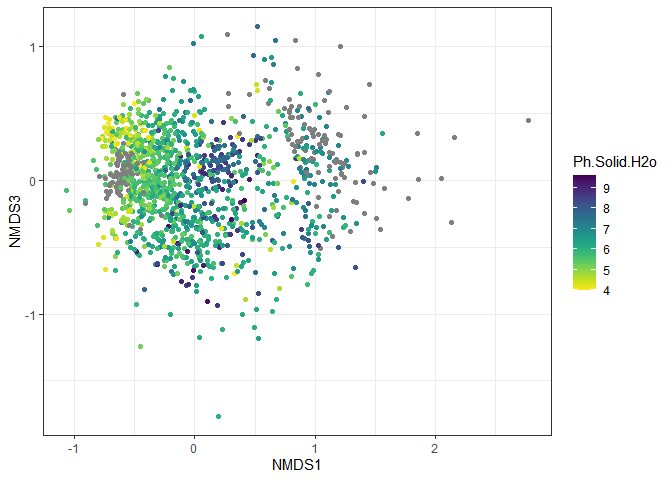
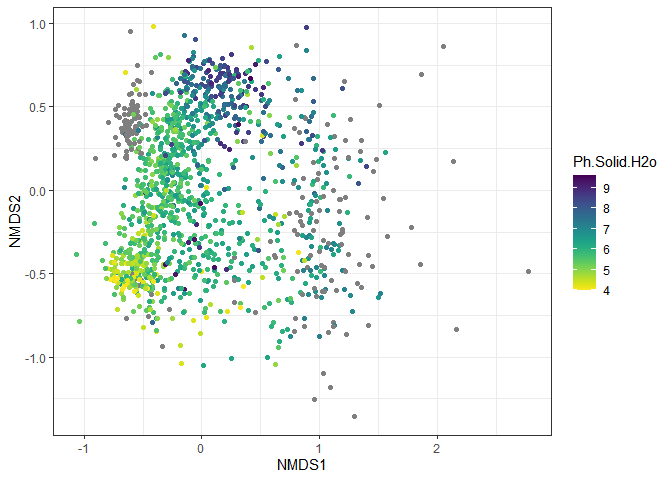
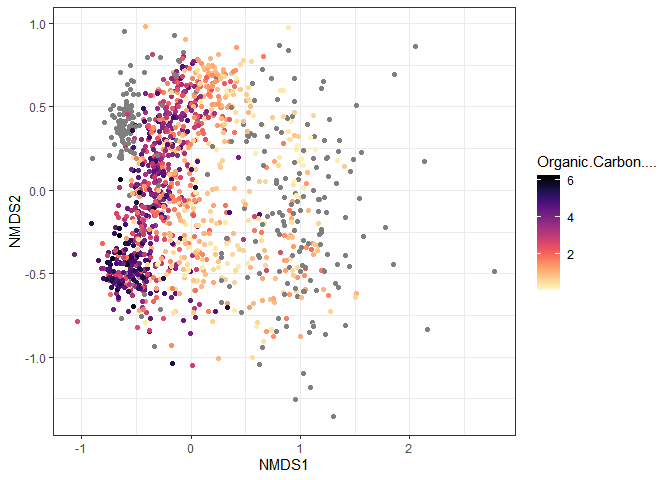
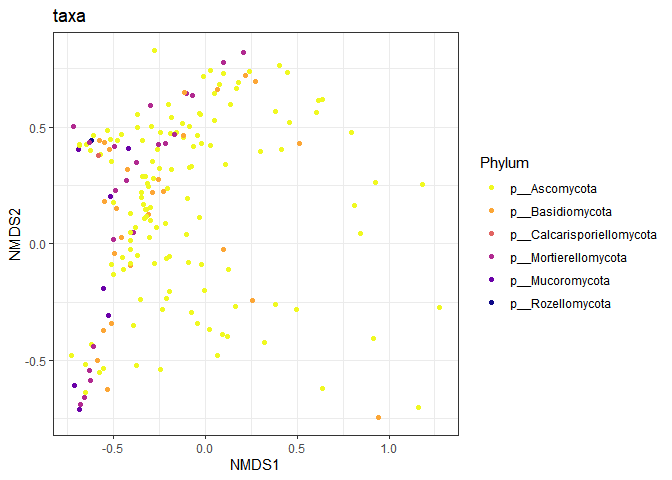
set.seed(1812)  
Fungi.Abun.NMDS = ordinate(abun.ps, "NMDS", "bray",   
 ### Bray-Curtis dissimilarity  
 k=3,   
 ### 3 dimensions  
 try=20, trymax=2000,   
 ### Min 20 random starts, max 2000  
 maxit=2000,   
 ### Try 5000 iterations at each start  
 smin = 0.05,   
 ### I would accept a stress of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

Fungi.Abun.NMDS

##   
## Call:  
## metaMDS(comm = veganifyOTU(physeq), distance = distance, k = 3, try = 20, trymax = 2000, trace = FALSE, plot = FALSE, maxit = 2000, smin = 0.05)   
##   
## global Multidimensional Scaling using monoMDS  
##   
## Data: veganifyOTU(physeq)   
## Distance: bray   
##   
## Dimensions: 3   
## Stress: 0.1581536   
## 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)'

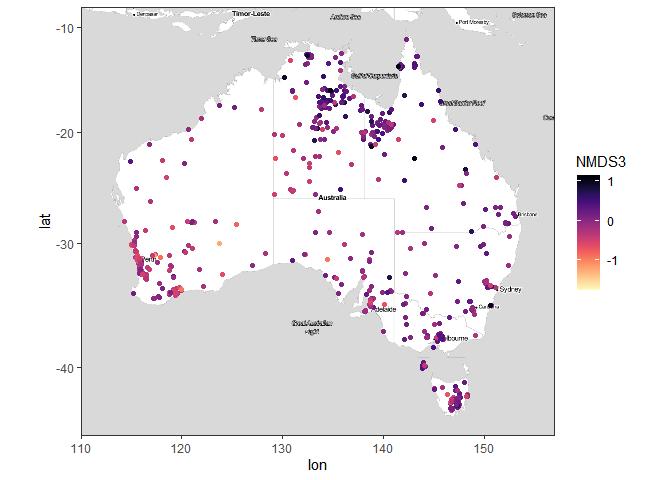
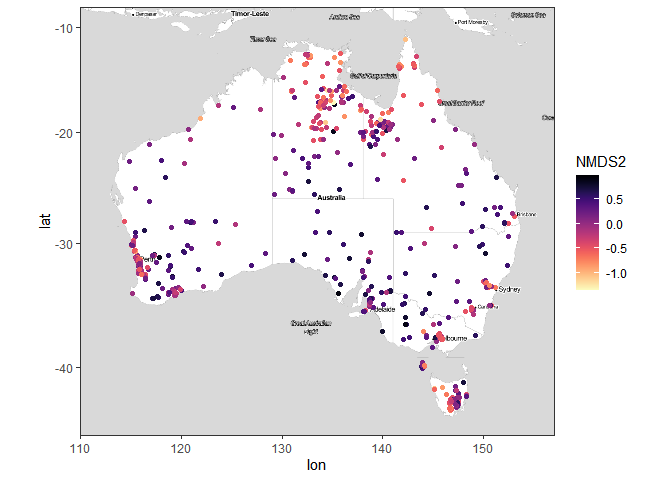
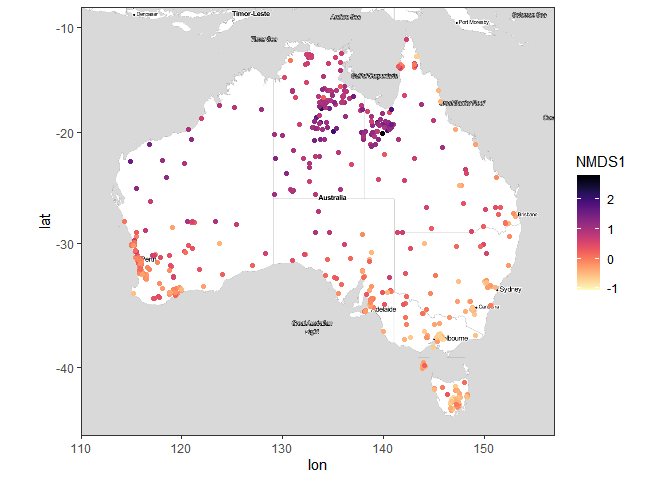
plot(Fungi.Abun.NMDS)





The stress of the NMDS is 0.158, suggesting that the structure of the data has been captured fairly.

The NMDS scores follow different spatial patterns. The first NMDS axis seem to follow a latitudinal as well as a coastal-inland direction, the second NMDS axis seems to follow the soil pH gradient



I save the NMDS scores into a separate dataframe with the context data, and it is saved in “fungi\_abun\_nmds\_scores.RData”.

## NMDS1 NMDS2 NMDS3 Latitude  
## SampleID.102.100.100/19259 -0.06024194 0.5899812 0.007961566 -35.63515  
## SampleID.102.100.100/42286 -0.20972740 0.2072062 0.028225060 -31.94138  
## SampleID.102.100.100/140258 -0.26513710 0.2005377 -0.033902713 -35.10306  
## SampleID.102.100.100/42608 0.97135867 -0.1842292 0.374475441 -16.59754  
## SampleID.102.100.100/138353 -0.09974202 0.6706748 0.439388678 -42.81674  
## SampleID.102.100.100/19487 -0.33958271 0.1302457 0.211275208 -33.60465  
## Longitude Depth\_Int  
## SampleID.102.100.100/19259 138.1184 0\_0.1  
## SampleID.102.100.100/42286 115.7739 0\_0.1  
## SampleID.102.100.100/140258 138.6849 0\_0.1  
## SampleID.102.100.100/42608 135.9827 0\_0.1  
## SampleID.102.100.100/138353 147.2502 0\_0.1  
## SampleID.102.100.100/19487 150.7304 0\_0.1

## Copulas models

Copulas are a flexible way to estimate the covariance matrix for species 1 … n in samples 1 … j Copulas can then be used to generated latent factors. These factors can akin to NDMS. Benefits - we generate the copulas from residuals of generalised linear models. This means we can fit any GLM to the data to fit its distribution and we can account for covariates or confounders in the model.

For the copulas we test negative binomial (NB) or zero-inflated negative binomial (ZINB) for abundance taxa. We selected the models with the minimum AIC – to keep it simple. For this approach we chose one model formula to use on every taxa – the model that has the lowest AIC across the greatest number of taxa.

For the functions of the ecoCopulas package, I extract the OTU table from the phyloseq object and transpose it, to have samples as rows and OTUs as columns.But the binomialnegative and the xero-inflated negative binomial work with integer data, so I bring the total abdundance phyloseq objects

InputDir <- "R:/PRJ-AusSoilMicrobes/Output/Fungi/"  
load(paste0(InputDir,"1\_AbsoluteAbunFungi.RData"))  
ps\_srf.r

## phyloseq-class experiment-level object  
## otu\_table() OTU Table: [ 60746 taxa and 1318 samples ]:  
## sample\_data() Sample Data: [ 1318 samples by 117 sample variables ]:  
## tax\_table() Taxonomy Table: [ 60746 taxa by 7 taxonomic ranks ]:  
## taxa are rows

Subset in the abundance data the taxa and samples present in the abundant-taxa phyloseq object

abunA.ps <- prune\_taxa(taxa\_names(ps\_srf.r)%in% taxa\_names(abun.ps),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: [ 184 taxa and 1317 samples ]:  
## sample\_data() Sample Data: [ 1317 samples by 117 sample variables ]:  
## tax\_table() Taxonomy Table: [ 184 taxa by 7 taxonomic ranks ]:  
## taxa are rows

I also create a factor indicating the origin of the data (BASE and GA), extract the context data into a separate dataframe and change the rownames to something more simple.

#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.792 OTU.925 OTU.1083 OTU.1724 OTU.1759 OTU.1872  
## SampleID.102.100.100/19259 0 0 0 0 4 0  
## SampleID.102.100.100/42286 0 0 0 0 0 0  
## SampleID.102.100.100/140258 1 0 0 10 0 0  
## SampleID.102.100.100/42608 0 3 0 0 0 0  
## SampleID.102.100.100/138353 0 0 1 0 1 0  
## SampleID.102.100.100/19487 0 0 0 0 0 0  
## SampleID.102.100.100/39143 0 0 0 0 0 0  
## SampleID.102.100.100/39177 0 0 0 0 2 0  
## SampleID.102.100.100/8270 10 0 16 0 0 0  
## SampleID.102.100.100/39322 35 0 0 0 0 0  
## OTU.1983 OTU.2339 OTU.3059 OTU.3329  
## SampleID.102.100.100/19259 17 24 0 0  
## SampleID.102.100.100/42286 0 672 0 5  
## SampleID.102.100.100/140258 0 44 0 0  
## SampleID.102.100.100/42608 0 0 0 0  
## SampleID.102.100.100/138353 0 0 18 7  
## SampleID.102.100.100/19487 31 302 0 0  
## SampleID.102.100.100/39143 38 24 12 3  
## SampleID.102.100.100/39177 1 15 12 0  
## SampleID.102.100.100/8270 4 0 0 0  
## SampleID.102.100.100/39322 0 30 0 0

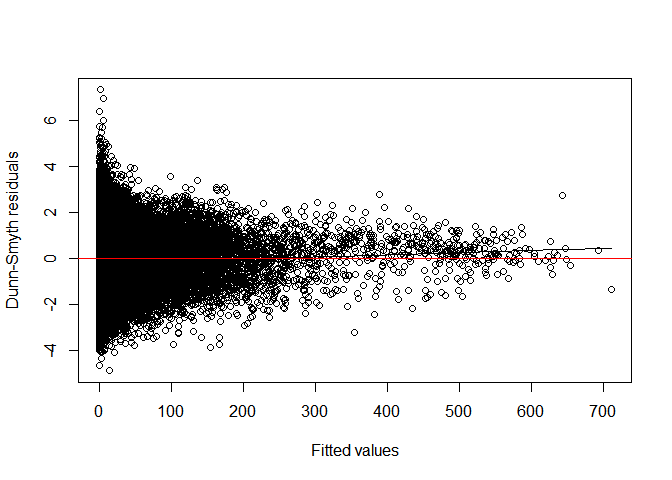
context.data <- data.frame(sample\_data(abunA.ps))  
context.data$Dataset <- ifelse(is.na(context.data$Ph.Solid.H2o), "GA", "BASE")  
rownames(context.data) <- 1:nrow(context.data)  
rownames(ta) <- 1:nrow(ta)

I fit models with the negative binomial distribution and relative abundance as response variable, including an offset with the number of sequences to transform to relative abundance. I test whether including the dataset of origin and depth interval improve the model.

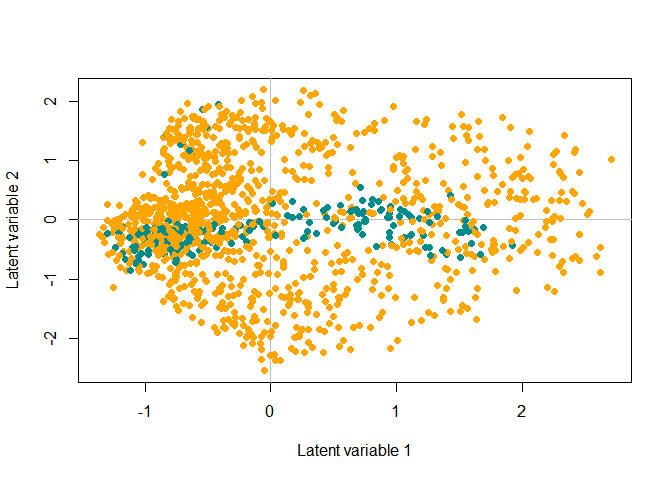
context.data$seqs<- log(rowSums(ta))  
  
sdm\_Nbinomial <-   
 stackedsdm(ta, formula\_X = ~1+ offset(seqs),  
 context.data,  
 family="negative.binomial", ncores = 7 )  
  
sdm\_Nbinomial\_2 <-   
 stackedsdm(ta, formula\_X = ~ 1 + Depth\_Int + Dataset+ offset(seqs),  
 context.data,  
 family="negative.binomial", ncores = 7 )  
gc()

## used (Mb) gc trigger (Mb) max used (Mb)  
## Ncells 8119957 433.7 11809212 630.7 11809212 630.7  
## Vcells 109514024 835.6 171790614 1310.7 133702172 1020.1

## plot residuals of models  
plot(sdm\_Nbinomial)



# Fit copula ordination   
abun\_lv=cord(sdm\_Nbinomial)



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.

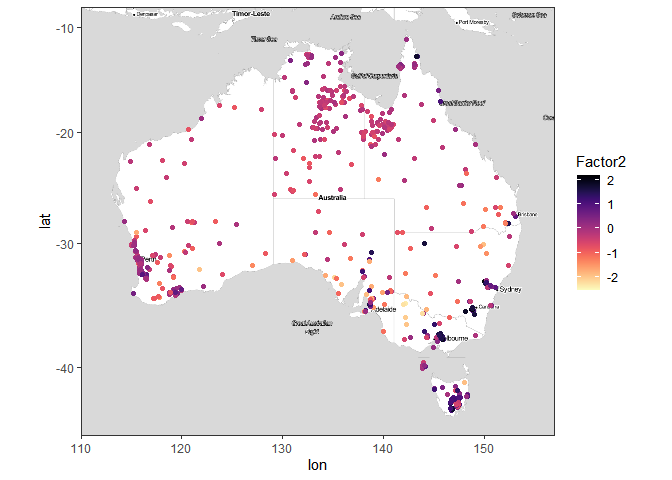
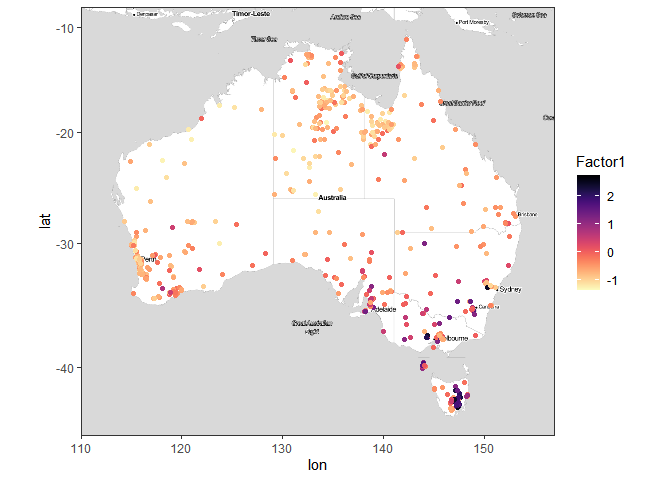
library(purrr)  
  
#max\_lv<- ncol(ra)-1   
  
lvs<- c(2,3,4,5)  
  
cords<-  
 map(lvs,  
 function(lv){  
 cord(sdm\_Nbinomial, nlv=lv)  
})  
  
best\_fit = map\_dbl(cords, ~ .x$BIC) %>% which.min()  
  
map\_dbl(cords, ~ .x$BIC)

## [1] -13263.73 -18085.05 -21716.25 -22775.64

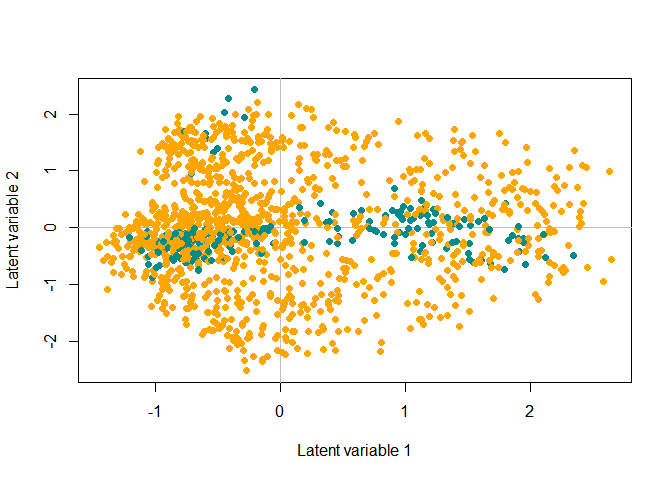
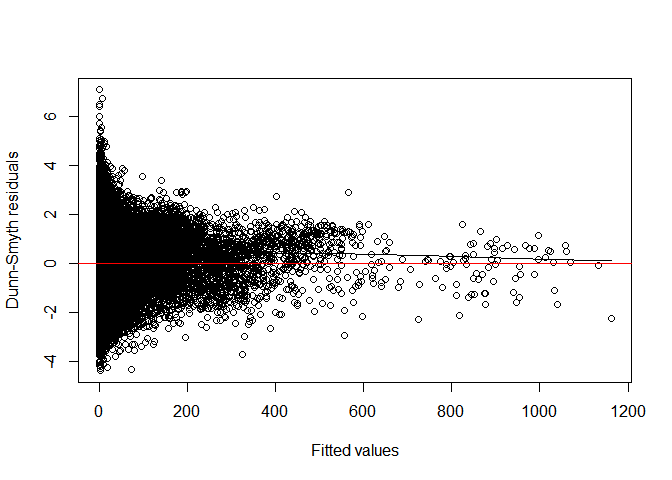
print(best\_fit)

## [1] 4

#### Visualize the latent factors



Now we check the residuals and ordination plot of the models fitted including depth and dataset as explanatory variables. The residuals indicate heterogeneity in variance when plotted against the fitted values, generally higher residuals for smaller fitted abundance.



Again, we check the number of optimal latent variables, as well as the BIC for the different ordinations

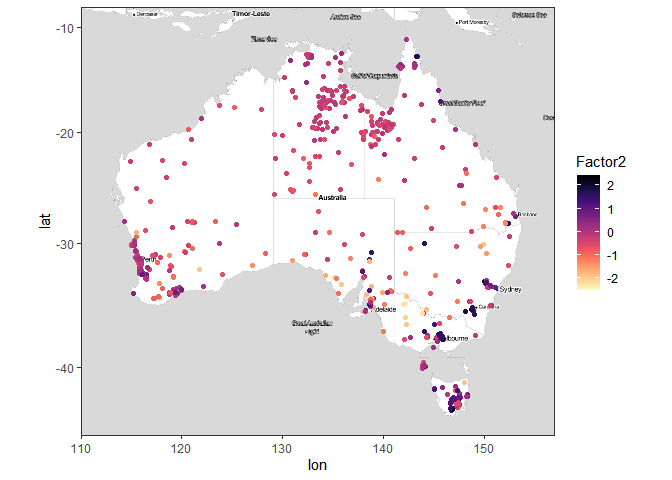
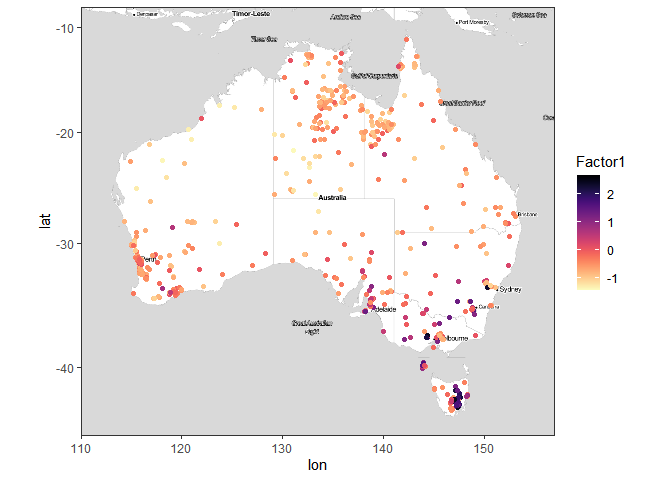
lvs<- c(2,3,4,5)  
  
cords<-  
 map(lvs,  
 function(lv){  
 cord(sdm\_Nbinomial\_2, nlv=lv)  
})  
  
best\_fit = map\_dbl(cords, ~ .x$BIC) %>% which.min()  
  
map\_dbl(cords, ~ .x$BIC)

## [1] -12727.06 -16789.22 -18485.95 -19129.94

print(best\_fit)

## [1] 4

And we plot the factors with their coordinates



#Save up to this point  
#save.image("C:/Users/mrom8073/OneDrive - The University of Sydney (Staff)/SoilMicrobialBiodiversity/Presentations/Fungi.Abun.Rdata.RData")

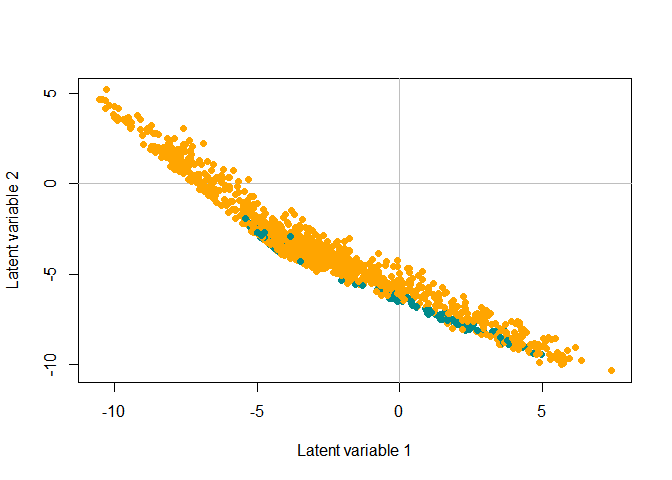
#### Zero-inflated negative binomial distribution

### Fit the ZINB models  
  
sdm\_Zi <-   
 stackedsdm(ta, formula\_X = ~1+ offset(seqs),  
 context.data,  
 family="zinegative.binomial", ncores = 7)  
  
sdm\_Zi\_2 <-   
 stackedsdm(ta, formula\_X = ~ 1 + Depth\_Int + Dataset + offset(seqs),  
 context.data,  
 family="zinegative.binomial", ncores = 7)

# Fit copula ordination   
library(countreg)  
abun\_lv=cord(sdm\_Zi\_2)

We plot the ordination plot of the zero-inflated negative binomial distribution controlling for depth and dataset or origin.

plot(abun\_lv, site.col=data.origin)

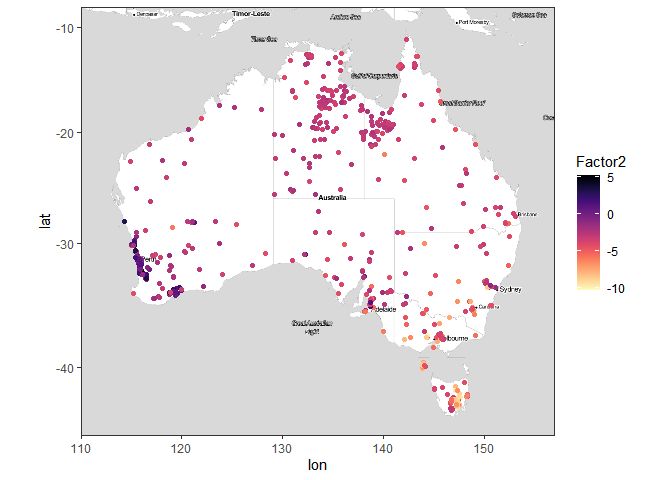
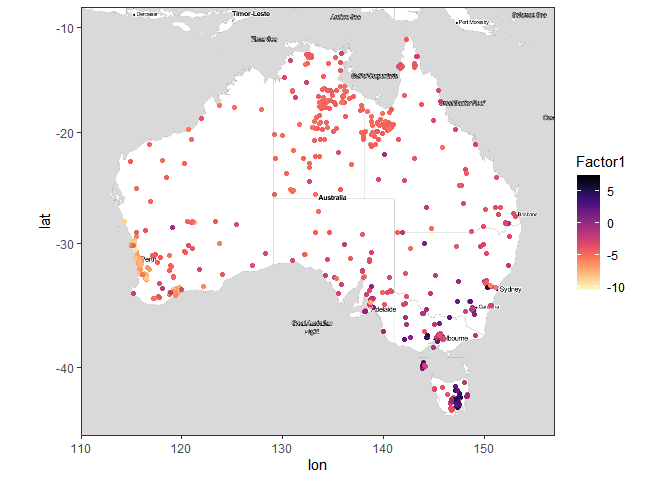


library(purrr)  
lvs<- c(2,3,4,5)  
  
cords<-  
 map(lvs,  
 function(lv){  
 cord(sdm\_Zi\_2,nlv=lv)  
})  
  
map\_dbl(cords, ~ .x$BIC)

## [1] -347517.7 -364840.9 -378531.0 -387968.6

best\_fit = map\_dbl(cords, ~ .x$BIC) %>% which.min()  
print(best\_fit)

## [1] 4



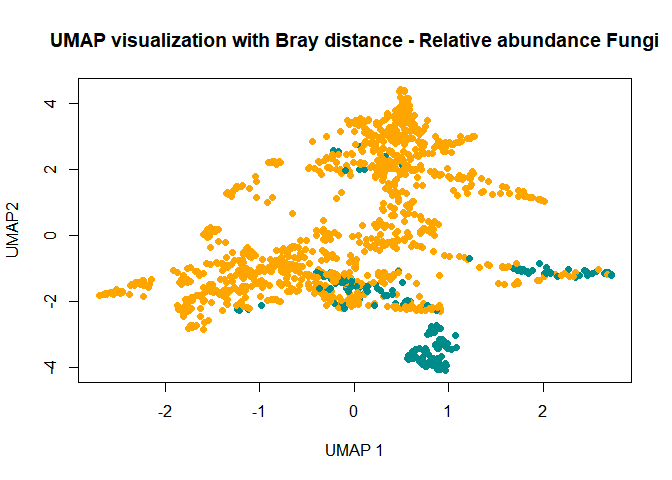
## used (Mb) gc trigger (Mb) max used (Mb)  
## Ncells 9007251 481.1 15099174 806.4 15099174 806.4  
## Vcells 119312705 910.3 567037728 4326.2 1320772586 10076.7

## UMAP

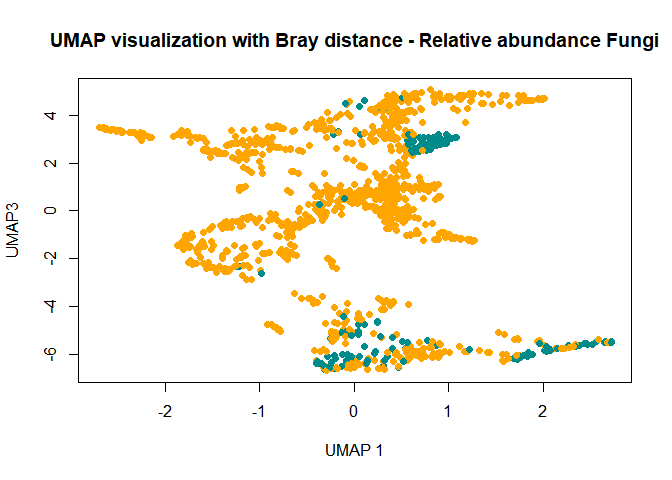
library(umap)  
### I calculate first Bray distances  
  
#Prepare the relative abundance in the right format  
ra <- abun.ps %>% otu\_table %>% as.matrix() %>% t() %>% as.data.frame()  
ra[1:10,1:10]

## OTU.792 OTU.925 OTU.1083 OTU.1724  
## SampleID.102.100.100/19259 0.0000000000 0.0000000000 0.0000000000 0.000000000  
## SampleID.102.100.100/42286 0.0000000000 0.0000000000 0.0000000000 0.000000000  
## SampleID.102.100.100/140258 0.0000999001 0.0000000000 0.0000000000 0.000999001  
## SampleID.102.100.100/42608 0.0000000000 0.0002997003 0.0000000000 0.000000000  
## SampleID.102.100.100/138353 0.0000000000 0.0000000000 0.0000999001 0.000000000  
## SampleID.102.100.100/19487 0.0000000000 0.0000000000 0.0000000000 0.000000000  
## SampleID.102.100.100/39143 0.0000000000 0.0000000000 0.0000000000 0.000000000  
## SampleID.102.100.100/39177 0.0000000000 0.0000000000 0.0000000000 0.000000000  
## SampleID.102.100.100/8270 0.0009990010 0.0000000000 0.0015984016 0.000000000  
## SampleID.102.100.100/39322 0.0034965035 0.0000000000 0.0000000000 0.000000000  
## OTU.1759 OTU.1872 OTU.1983 OTU.2339  
## SampleID.102.100.100/19259 0.0003996004 0 0.0016983017 0.002397602  
## SampleID.102.100.100/42286 0.0000000000 0 0.0000000000 0.067132867  
## SampleID.102.100.100/140258 0.0000000000 0 0.0000000000 0.004395604  
## SampleID.102.100.100/42608 0.0000000000 0 0.0000000000 0.000000000  
## SampleID.102.100.100/138353 0.0000999001 0 0.0000000000 0.000000000  
## SampleID.102.100.100/19487 0.0000000000 0 0.0030969031 0.030169830  
## SampleID.102.100.100/39143 0.0000000000 0 0.0037962038 0.002397602  
## SampleID.102.100.100/39177 0.0001998002 0 0.0000999001 0.001498501  
## SampleID.102.100.100/8270 0.0000000000 0 0.0003996004 0.000000000  
## SampleID.102.100.100/39322 0.0000000000 0 0.0000000000 0.002997003  
## OTU.3059 OTU.3329  
## SampleID.102.100.100/19259 0.000000000 0.0000000000  
## SampleID.102.100.100/42286 0.000000000 0.0004995005  
## SampleID.102.100.100/140258 0.000000000 0.0000000000  
## SampleID.102.100.100/42608 0.000000000 0.0000000000  
## SampleID.102.100.100/138353 0.001798202 0.0006993007  
## SampleID.102.100.100/19487 0.000000000 0.0000000000  
## SampleID.102.100.100/39143 0.001198801 0.0002997003  
## SampleID.102.100.100/39177 0.001198801 0.0000000000  
## SampleID.102.100.100/8270 0.000000000 0.0000000000  
## SampleID.102.100.100/39322 0.000000000 0.0000000000

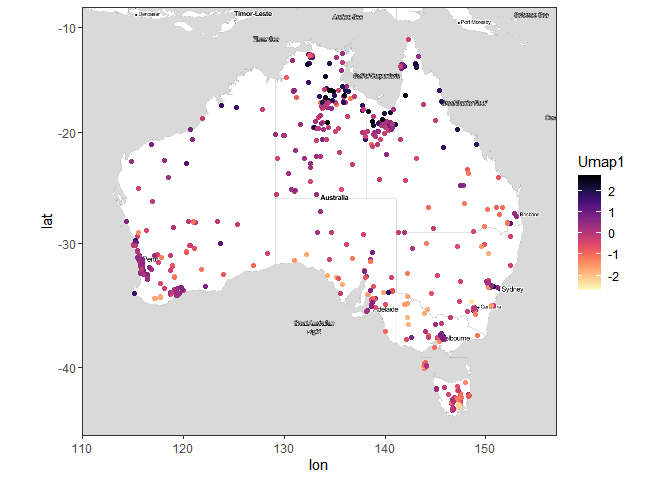
bray.dist <- vegan::vegdist(ra, "bray")  
  
### Tune parameters  
custom.config = umap.defaults  
custom.config$n\_components=3  
custom.config$random\_state = 1984  
  
### Perform UMAP  
fungi.umap = umap(as.matrix(bray.dist),config=custom.config, input="dist")  
  
### Plot  
umap.context <- sample\_data(abun.ps)  
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$Umap1 <- fungi.umap$layout[,1]  
umap.context$Umap2 <- fungi.umap$layout[,2]  
umap.context$Umap3 <- fungi.umap$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 Bray distance - Relative abundance Fungi")



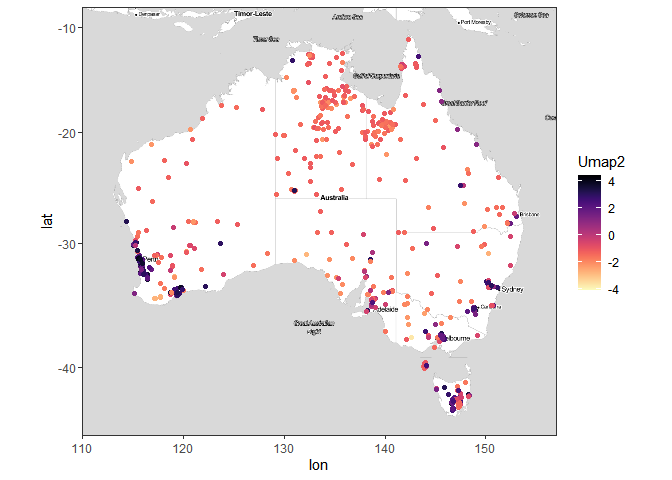
plot(x=umap.context$Umap1, y=umap.context$Umap3 , col=data.origin, pch=19, xlab="UMAP 1", ylab="UMAP3",  
 main="UMAP visualization with Bray distance - Relative abundance Fungi")



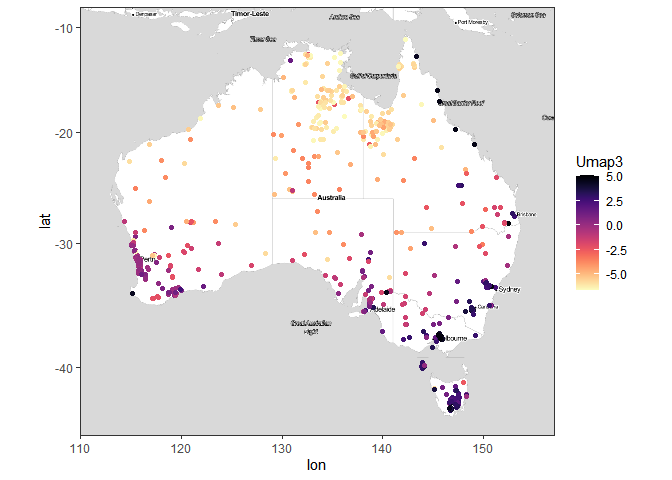
ggmap(AusMap) + geom\_point(aes(y = Latitude, x =Longitude,color=Umap1), data = umap.context) +  
 scale\_color\_viridis(discrete = FALSE, option="A", direction = -1)



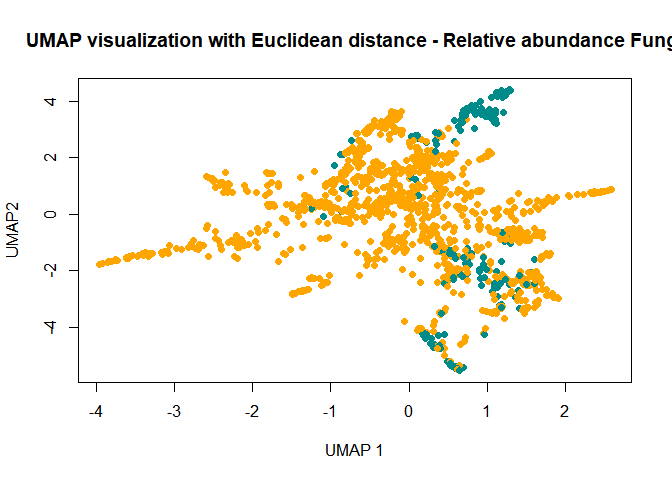
ggmap(AusMap) + geom\_point(aes(y = Latitude, x =Longitude,color=Umap2), data = umap.context) +  
 scale\_color\_viridis(discrete = FALSE, option="A", direction = -1)



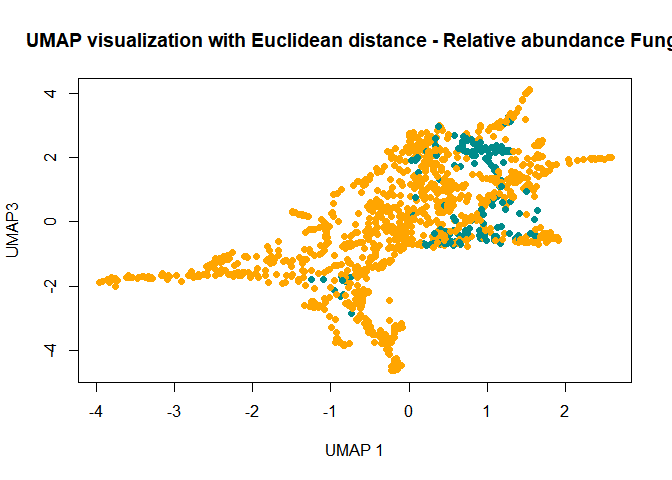
ggmap(AusMap) + geom\_point(aes(y = Latitude, x =Longitude,color=Umap3), data = umap.context) +  
 scale\_color\_viridis(discrete = FALSE, option="A", direction = -1)



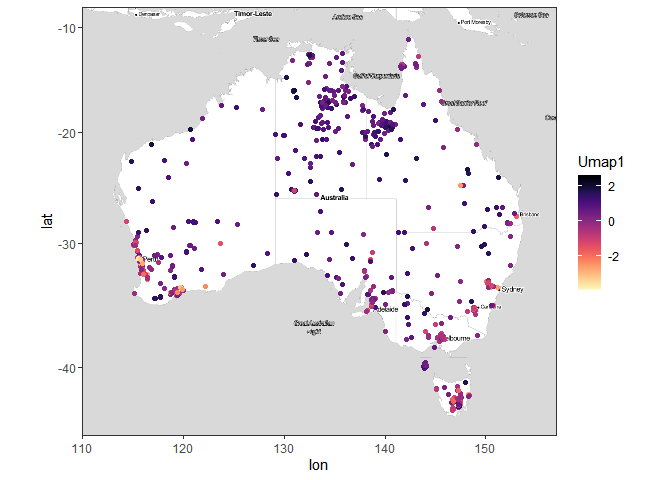
### Tune parameters  
custom.config = umap.defaults  
custom.config$n\_components=3  
custom.config$random\_state = 1984  
  
fungi.umap2 = umap(ra, config=custom.config)  
  
### Plot  
umap.context <- sample\_data(abun.ps)  
umap.context$Dataset <- ifelse(is.na(umap.context$Ph.Solid.H2o), "GA", "BASE")  
umap.context$Dataset <- as.factor(umap.context$Dataset)  
  
### Attach the umap scores  
umap.context$Umap1 <- fungi.umap2$layout[,1]  
umap.context$Umap2 <- fungi.umap2$layout[,2]  
umap.context$Umap3 <- fungi.umap2$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 Fungi")



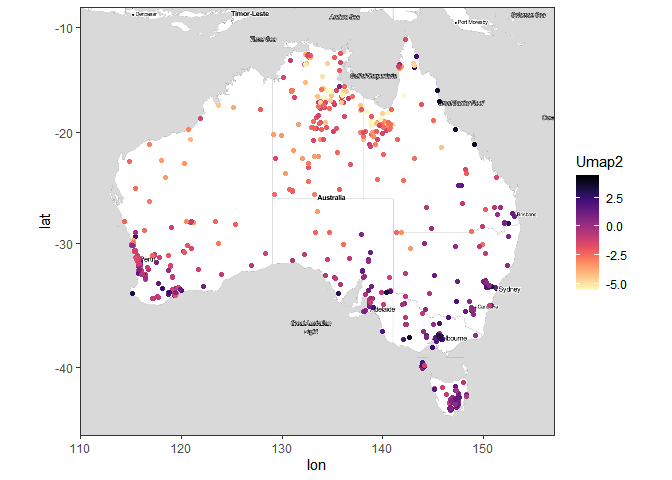
plot(x=umap.context$Umap1, y=umap.context$Umap3 , col=data.origin, pch=19, xlab="UMAP 1", ylab="UMAP3",  
 main="UMAP visualization with Euclidean distance - Relative abundance Fungi")



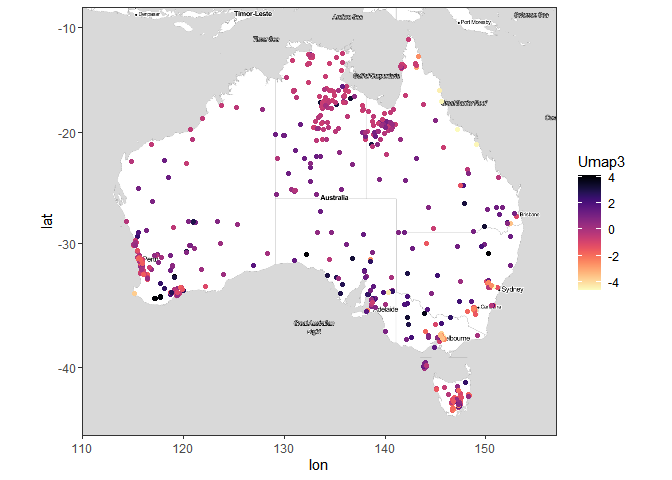
ggmap(AusMap) + geom\_point(aes(y = Latitude, x =Longitude,color=Umap1), data = umap.context) +  
 scale\_color\_viridis(discrete = FALSE, option="A", direction = -1)



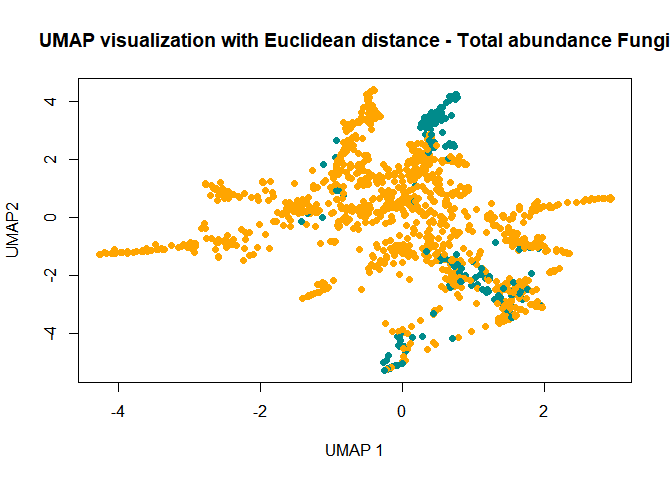
ggmap(AusMap) + geom\_point(aes(y = Latitude, x =Longitude,color=Umap2), data = umap.context) +  
 scale\_color\_viridis(discrete = FALSE, option="A", direction = -1)



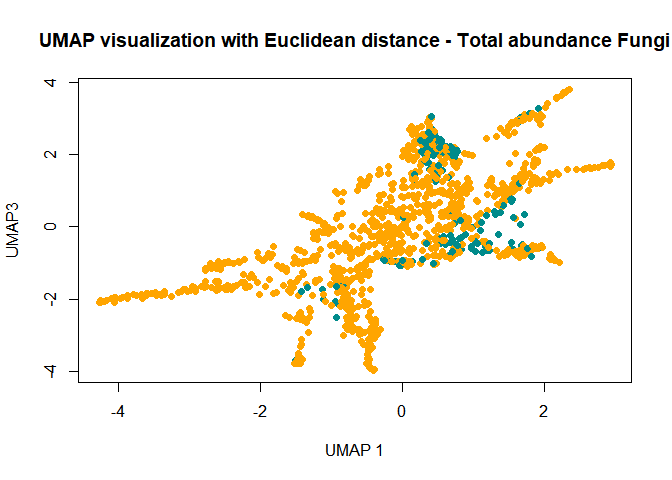
ggmap(AusMap) + geom\_point(aes(y = Latitude, x =Longitude,color=Umap3), data = umap.context) +  
 scale\_color\_viridis(discrete = FALSE, option="A", direction = -1)

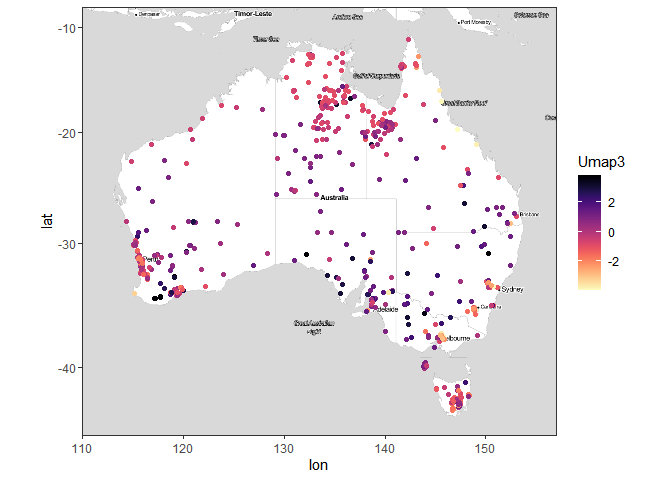
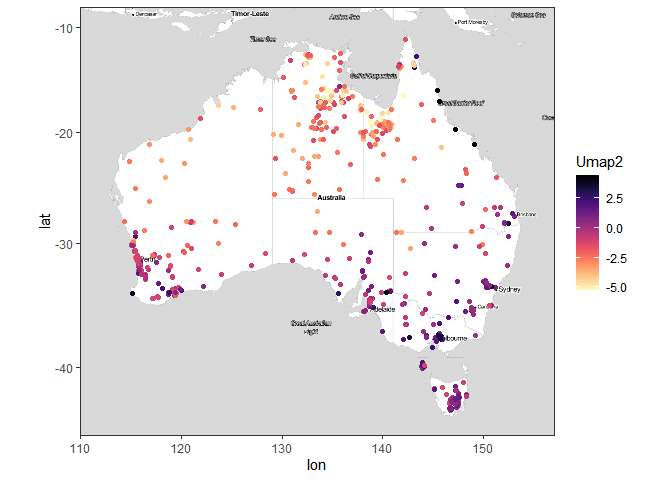
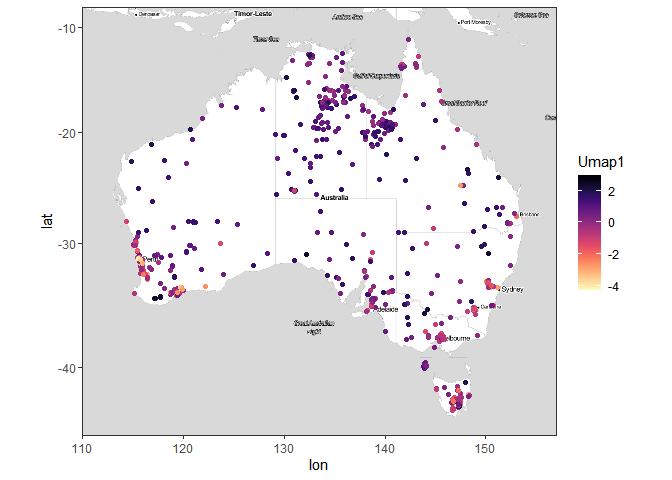


### Tune parameters  
custom.config = umap.defaults  
custom.config$n\_components=3  
custom.config$random\_state = 1984  
  
fungi.umap3 = umap(ta,config=custom.config)  
  
### Plot  
umap.context <- sample\_data(abun.ps)  
umap.context$Dataset <- ifelse(is.na(umap.context$Ph.Solid.H2o), "GA", "BASE")  
umap.context$Dataset <- as.factor(umap.context$Dataset)  
  
### Attach the umap scores  
umap.context$Umap1 <- fungi.umap3$layout[,1]  
umap.context$Umap2 <- fungi.umap3$layout[,2]  
umap.context$Umap3 <- fungi.umap3$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 - Total abundance Fungi")



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 - Total abundance Fungi")





save.image("C:/Users/mrom8073/OneDrive - The University of Sydney (Staff)/SoilMicrobialBiodiversity/Presentations/Fungi.Abun.3Analyses.RData")  
# load("C:/Users/mrom8073/OneDrive - The University of Sydney (Staff)/SoilMicrobialBiodiversity/Presentations/Fungi.Abun.3Analyses.RData")