MICB 405 Project 2

Team 17

November 17, 2018

### Load in Themes

#Before we get started, I am going to load a function to create a theme from https://rpubs.com/Koundy/71792, which is a nice theme that I will use throughout the rest of the plots.  
  
theme\_Publication <- function(base\_size=12) {  
 (theme\_foundation(base\_size=base\_size)  
 + theme(plot.title = element\_text(face = "bold",  
 size = rel(1.2), hjust = 0.5),  
 text = element\_text(),  
 panel.background = element\_rect(colour = NA),  
 plot.background = element\_rect(colour = NA),  
 panel.border = element\_rect(colour = NA),  
 axis.title = element\_text(face = "bold"), #,size = rel(1)  
 axis.title.y = element\_text(angle=90,vjust =2),  
 axis.title.x = element\_text(vjust = -0.2),  
 axis.text = element\_text(),   
 axis.line = element\_line(colour="black"),  
 axis.ticks = element\_line(),  
 panel.grid.major = element\_line(colour="#f0f0f0"),  
 panel.grid.minor = element\_blank(),  
 legend.key = element\_rect(colour = NA),  
 legend.position = "bottom",  
 legend.direction = "horizontal",  
 legend.key.size= unit(0.2, "cm"),  
 legend.margin = unit(0, "cm"),  
 legend.title = element\_text(face="italic"),  
 plot.margin=unit(c(10,5,5,5),"mm"),  
 strip.background=element\_rect(colour="#f0f0f0",fill="#f0f0f0"),  
 strip.text = element\_text(face="bold")  
 ))  
   
}  
  
theme\_Publication\_legend\_side <- function(base\_size=12) {  
 (theme\_foundation(base\_size=base\_size)  
 + theme(plot.title = element\_text(face = "bold",  
 size = rel(1.2), hjust = 0.5),  
 text = element\_text(),  
 panel.background = element\_rect(colour = NA),  
 plot.background = element\_rect(colour = NA),  
 panel.border = element\_rect(colour = NA),  
 axis.title = element\_text(face = "bold"), #,size = rel(1)  
 axis.title.y = element\_text(angle=90,vjust =2),  
 axis.title.x = element\_text(vjust = -0.2),  
 axis.text = element\_text(),   
 axis.line = element\_line(colour="black"),  
 axis.ticks = element\_line(),  
 panel.grid.major = element\_line(colour="#f0f0f0"),  
 panel.grid.minor = element\_blank(),  
 legend.key = element\_rect(colour = NA),  
 legend.key.size= unit(0.2, "cm"),  
 legend.margin = unit(0, "cm"),  
 legend.title = element\_text(face="italic"),  
 plot.margin=unit(c(10,5,5,5),"mm"),  
 strip.background=element\_rect(colour="#f0f0f0",fill="#f0f0f0"),  
 strip.text = element\_text(face="bold")  
 ))  
   
}  
  
theme\_Publication\_facets <- function(base\_size=12) {  
 (theme\_foundation(base\_size=base\_size)  
 + theme(plot.title = element\_text(face = "bold",  
 size = rel(1.2), hjust = 0.5),  
 text = element\_text(),  
 panel.background = element\_rect(colour = NA),  
 plot.background = element\_rect(colour = NA),  
 axis.title = element\_text(face = "bold"), #,size = rel(1)  
 axis.title.y = element\_text(angle=90,vjust =2),  
 axis.title.x = element\_text(vjust = -0.2),  
 axis.text = element\_text(),   
 axis.line = element\_line(colour="black"),  
 axis.ticks = element\_line(),  
 panel.grid.major = element\_line(colour="#f0f0f0"),  
 panel.grid.minor = element\_blank(),  
 legend.key = element\_rect(colour = NA),  
 legend.position = "bottom",  
 legend.direction = "horizontal",  
 legend.key.size= unit(0.2, "cm"),  
 legend.margin = unit(0, "cm"),  
 legend.title = element\_text(face="italic"),  
 plot.margin=unit(c(10,5,5,5),"mm"),  
 strip.background=element\_rect(colour="#f0f0f0",fill="#f0f0f0"),  
 strip.text = element\_text(face="bold")  
 ))  
   
}

## Data

Load the data and put it in a format that we need for plotting

#--- GETTING THE DATA ----  
  
########################  
# ---- Saanich Data ----  
# Load raw data  
raw\_dat <- readr::read\_csv("Saanich\_Data.csv")  
  
#Clean dat  
dat <-   
 raw\_dat %>%  
 dplyr::filter(!is.na(WS\_O2)) %>%  
 dplyr::rename(O2\_uM=WS\_O2, NO3\_uM=WS\_NO3, H2S\_uM=WS\_H2S) %>%  
 dplyr::mutate(Depth\_m=Depth\*1000)  
  
########################  
# ---- checkM data ----  
checkm\_dat <- read.table("MetaBAT2\_SaanichInlet\_200m\_min1500\_checkM\_stdout.tsv",  
 header=TRUE,  
 sep="\t",  
 comment.char = '') %>%   
 dplyr::rename(mag = Bin.Id) %>%   
 dplyr::select(mag, Completeness, Contamination)  
  
# Due to a bug in the renaming script we have to rename the bins. Its a bit hacky but works using tidyverse functions  
metag\_rpkm <- read.table("SaanichInlet\_200m\_binned.rpkm.csv", header=T, sep=',') %>%   
 mutate(Sequence = gsub('m\_', 'm.', Sequence)) %>%   
 mutate(Sequence = gsub('Inlet\_', 'Inlet.', Sequence)) %>%   
 separate(col=Sequence, into=c("mag", "contig"), sep='\_', extra="merge") %>%   
 group\_by(Sample, mag) %>%   
 summarise(g\_rpkm = mean(RPKM)) %>%   
 mutate(mag = gsub('Inlet.', 'Inlet\_', mag))   
 # dplyr::filter(Sample != "SI042\_200m")  
  
########################  
# ---- rpkm and mag data ----  
#Read in KO Annotation tables  
ko <- read.table("SaanichInlet\_MAGs\_ORFs\_ko.cleaned.txt") %>%   
 dplyr::rename(orf = V1) %>%   
 dplyr::rename(ko = V2)  
  
#Read in all rpkm files from each cruise in August  
rpkm\_48 <- read.table("SI048\_200m.RPKM.csv", header=FALSE, sep=',') %>%   
 dplyr::rename(orf = V1) %>%   
 dplyr::rename(rpkm = V2)  
rpkm\_72 <- read.table("SI072\_200m.RPKM.csv", header=FALSE, sep=',') %>%   
 dplyr::rename(orf = V1) %>%   
 dplyr::rename(rpkm = V2)  
rpkm\_73 <- read.table("SI073\_200m.RPKM.csv", header=FALSE, sep=',') %>%   
 dplyr::rename(orf = V1) %>%   
 dplyr::rename(rpkm = V2)  
rpkm\_74 <- read.table("SI074\_200m.RPKM.csv", header=FALSE, sep=',') %>%   
 dplyr::rename(orf = V1) %>%   
 dplyr::rename(rpkm = V2)  
rpkm\_75 <- read.table("SI075\_200m.RPKM.csv", header=FALSE, sep=',') %>%   
 dplyr::rename(orf = V1) %>%   
 dplyr::rename(rpkm = V2)  
  
#Read in prokka MAG map  
prokka\_mag\_map <- read.table("Prokka\_MAG\_map\_basename.csv", header=F, sep=',') %>%   
 dplyr::rename(prokka\_id = V1) %>%   
 dplyr::rename(mag = V2)  
  
#Read in gtdbtk files  
arc\_class <- read.table("gtdbtk.ar122.classification\_pplacer.tsv", sep="\t")  
bac\_class <- read.table("gtdbtk.bac120.classification\_pplacer.tsv", sep="\t")  
  
#Combine archaea and bacteria gtdb files  
gtdb\_dat <- rbind(arc\_class, bac\_class) %>%   
 dplyr::rename(mag = V1) %>%   
 separate(V2, sep=';', into=c("Kingdom", "Phylum", "Class", "Order", "Family", "Genus", "Species"))  
  
#We can also determine the number of Class present in our bins, and how many MAGs are representing each.  
gtdb\_dat %>%   
 group\_by(Order) %>%   
 summarise(count = n\_distinct(mag)) %>%   
 knitr::kable()

|  |  |
| --- | --- |
| Order | count |
| o\_\_ | 7 |
| o\_\_Anaerolineales | 2 |
| o\_\_Babeliales | 1 |
| o\_\_Bacteroidales | 2 |
| o\_\_BD1-5 | 1 |
| o\_\_Berkiellales | 1 |
| o\_\_Betaproteobacteriales | 1 |
| o\_\_Elusimicrobiales | 1 |
| o\_\_Flavobacteriales | 2 |
| o\_\_GW2011-AR9 | 2 |
| o\_\_Iainarchaeales | 1 |
| o\_\_Komeilibacterales | 3 |
| o\_\_Magasanikibacterales | 1 |
| o\_\_Marinisomatales | 1 |
| o\_\_Methylococcales | 1 |
| o\_\_Moranbacterales | 1 |
| o\_\_Nitrososphaerales | 2 |
| o\_\_Pacearchaeales | 3 |
| o\_\_Peregrinibacterales | 1 |
| o\_\_Pseudomonadales | 3 |
| o\_\_Rhodobacterales | 1 |
| o\_\_SCGC-AAA011-G17 | 3 |
| o\_\_SG8-23 | 1 |
| o\_\_Sneathiellales | 1 |
| o\_\_TCS64 | 1 |
| o\_\_Thiomicrospirales | 1 |
| o\_\_UBA12501 | 1 |
| o\_\_UBA1400 | 1 |
| o\_\_UBA6615 | 1 |
| o\_\_UBA7916 | 1 |
| o\_\_XYD2-FULL-50-16 | 1 |

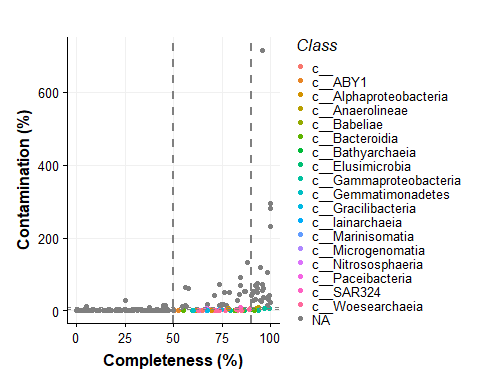
gtdb\_dat <- dplyr::select(gtdb\_dat, mag, Kingdom, Phylum, Class, Order, Family)  
  
#combine all cruises and ko values  
ko\_rpkm\_48 <- left\_join(ko, rpkm\_48, by="orf") %>%   
 separate(orf, into=c("prokka\_id", "orf\_id")) %>% # Split the Prokka ORF names into MAG identifier and ORF  
 left\_join(prokka\_mag\_map, by="prokka\_id") %>%   
 left\_join(gtdb\_dat, by="mag") %>%  
 left\_join(checkm\_dat, by="mag") %>%  
 left\_join(metag\_rpkm, by="mag")  
ko\_rpkm\_72 <- left\_join(ko, rpkm\_72, by="orf") %>%   
 separate(orf, into=c("prokka\_id", "orf\_id")) %>% # Split the Prokka ORF names into MAG identifier and ORF  
 left\_join(prokka\_mag\_map, by="prokka\_id") %>%   
 left\_join(gtdb\_dat, by="mag")%>%  
 left\_join(checkm\_dat, by="mag") %>%  
 left\_join(metag\_rpkm, by="mag")  
ko\_rpkm\_73 <- left\_join(ko, rpkm\_73, by="orf") %>%   
 separate(orf, into=c("prokka\_id", "orf\_id")) %>% # Split the Prokka ORF names into MAG identifier and ORF  
 left\_join(prokka\_mag\_map, by="prokka\_id") %>%   
 left\_join(gtdb\_dat, by="mag")%>%  
 left\_join(checkm\_dat, by="mag") %>%  
 left\_join(metag\_rpkm, by="mag")  
ko\_rpkm\_74 <- left\_join(ko, rpkm\_74, by="orf") %>%   
 separate(orf, into=c("prokka\_id", "orf\_id")) %>% # Split the Prokka ORF names into MAG identifier and ORF  
 left\_join(prokka\_mag\_map, by="prokka\_id") %>%   
 left\_join(gtdb\_dat, by="mag")%>%  
 left\_join(checkm\_dat, by="mag") %>%  
 left\_join(metag\_rpkm, by="mag")  
ko\_rpkm\_75 <- left\_join(ko, rpkm\_75, by="orf") %>%   
 separate(orf, into=c("prokka\_id", "orf\_id")) %>% # Split the Prokka ORF names into MAG identifier and ORF  
 left\_join(prokka\_mag\_map, by="prokka\_id") %>%   
 left\_join(gtdb\_dat, by="mag")%>%  
 left\_join(checkm\_dat, by="mag") %>%  
 left\_join(metag\_rpkm, by="mag")  
  
#combine all ko\_rpkm data into one data frame  
rpkm\_dat <- rbind(ko\_rpkm\_48, ko\_rpkm\_73, ko\_rpkm\_74, ko\_rpkm\_75)  
head(rpkm\_dat) %>% knitr::kable()

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| prokka\_id | orf\_id | ko | rpkm | mag | Kingdom | Phylum | Class | Order | Family | Completeness | Contamination | Sample | g\_rpkm |
| EBFFFBAL | 00003 | K07478 | 0 | SaanichInlet\_200m.107 | d\_\_Bacteria | p\_\_Chloroflexota | c\_\_Anaerolineae | o\_\_Anaerolineales | f\_\_ | 92.73 | 6.36 | SI042\_200m | 1.5948984 |
| EBFFFBAL | 00003 | K07478 | 0 | SaanichInlet\_200m.107 | d\_\_Bacteria | p\_\_Chloroflexota | c\_\_Anaerolineae | o\_\_Anaerolineales | f\_\_ | 92.73 | 6.36 | SI048\_200m | 0.8125208 |
| EBFFFBAL | 00003 | K07478 | 0 | SaanichInlet\_200m.107 | d\_\_Bacteria | p\_\_Chloroflexota | c\_\_Anaerolineae | o\_\_Anaerolineales | f\_\_ | 92.73 | 6.36 | SI072\_200m | 0.3522020 |
| EBFFFBAL | 00003 | K07478 | 0 | SaanichInlet\_200m.107 | d\_\_Bacteria | p\_\_Chloroflexota | c\_\_Anaerolineae | o\_\_Anaerolineales | f\_\_ | 92.73 | 6.36 | SI073\_200m | 0.2647460 |
| EBFFFBAL | 00003 | K07478 | 0 | SaanichInlet\_200m.107 | d\_\_Bacteria | p\_\_Chloroflexota | c\_\_Anaerolineae | o\_\_Anaerolineales | f\_\_ | 92.73 | 6.36 | SI074\_200m | 0.1810810 |
| EBFFFBAL | 00003 | K07478 | 0 | SaanichInlet\_200m.107 | d\_\_Bacteria | p\_\_Chloroflexota | c\_\_Anaerolineae | o\_\_Anaerolineales | f\_\_ | 92.73 | 6.36 | SI075\_200m | 0.2680613 |

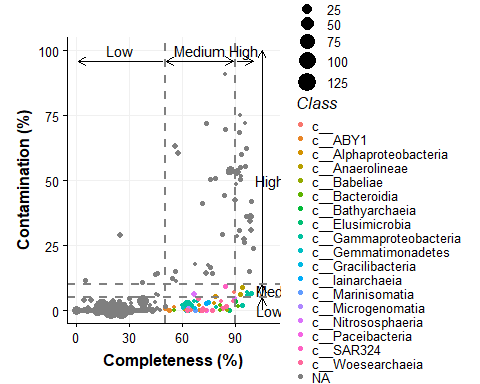
## Contamination vs Completeness Scatter Plot

First we will analyse the MAGs from checkM output with rpkm abundance values Completeness and contamination ranges are based on quality standard definition from Bowers et al., 2017

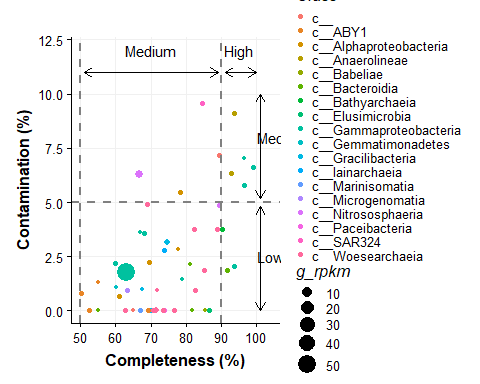
rpkm\_dat <- left\_join(metag\_rpkm, checkm\_dat, by="mag") %>%   
 left\_join(gtdb\_dat, by="mag") %>%   
 group\_by(mag, Kingdom, Phylum, Class, Completeness, Contamination) %>%   
 summarise(g\_rpkm = mean(g\_rpkm))  
  
#-- PLOT WITH ALL OF THE DATA! --  
#Plotting Completeness vs Contamination of MAGs at 200m  
rpkm\_dat %>%  
 ggplot(aes(x=Completeness, y=Contamination, col=Class)) +  
 geom\_point() +  
 geom\_vline(xintercept = 50, linetype = "dashed", size = 1, color = "#808080") + #med completeness quality  
 geom\_vline(xintercept = 90, linetype = "dashed", size = 1, color = "#808080") + #high completeness quality  
 geom\_hline(yintercept = 5, linetype = "dashed", size = 0.65, color = "#808080") + #med contamination quality = 5-10%  
 geom\_hline(yintercept = 10, linetype = "dashed", size = 0.65, color = "#808080") + #high contamination quality = > 10%  
 xlab("Completeness (%)") +  
 ylab("Contamination (%)") +  
 theme\_Publication\_legend\_side()



#-- PLOT WITH CONTAMINATION <= 100%! --  
#Plotting completeness vs contamination of MAGS at 200m, where contamination <=100%  
# Note: contamination above 100% indicates that the recovered bin likely contains multiple organisms. To analyze closer, this plot removes the contamination that is > 100%.  
rpkm\_dat %>%  
 dplyr::filter(Contamination < 100) %>%  
 ggplot(aes(x=Completeness, y=Contamination, colour=Class)) +  
 geom\_point(aes(size=g\_rpkm)) +  
 #-- LOW COMPLETENESS--  
 annotate("text", x=25, y=100, label="Low",   
 colour="black", text=element\_text(size=9, family="Arial")) +  
 geom\_segment(aes(x = 49, y = 96, xend = 1, yend = 96),  
 arrow = arrow(length = unit(0.25, "cm")), colour="black")+  
 #-- MED COMPLETENESS--  
 geom\_vline(xintercept = 50, linetype = "dashed", size = 1, color = "#808080") + #med completeness quality  
 annotate("text", x=70, y=100, label="Medium",   
 colour="black", text=element\_text(size=9, family="Arial")) +  
 geom\_segment(aes(x = 51, y = 96, xend = 89, yend = 96),  
 arrow = arrow(length = unit(0.25, "cm")), colour="black")+  
 geom\_segment(aes(x = 89, y = 96, xend = 51, yend = 96),  
 arrow = arrow(length = unit(0.25, "cm")), colour="black")+  
 #-- HIGH COMPLETENESS --  
 geom\_vline(xintercept = 90, linetype = "dashed", size = 1, color = "#808080") + #high completeness quality  
 annotate("text", x=95, y=100, label="High",   
 colour="black", text=element\_text(size=9, family="Arial")) +  
 geom\_segment(aes(x = 91, y = 96, xend = 100, yend = 96),  
 arrow = arrow(length = unit(0.25, "cm")), colour="black")+  
 #-- LOW CONTAMINATION --  
 annotate("text", x=110, y=0, label="Low", #angle=270,  
 colour="black", text=element\_text(size=9, family="Arial")) +  
 geom\_segment(aes(x = 105, y = 0, xend = 105, yend = 5),  
 arrow = arrow(length = unit(0.25, "cm")), colour="black")+  
 #-- MED CONTAMINATION --  
 annotate("text", x=110, y=8, label="Med", #angle=270,  
 colour="black", text=element\_text(size=9, family="Arial")) +  
 geom\_segment(aes(x = 105, y = 5, xend = 105, yend = 10),  
 arrow = arrow(length = unit(0.25, "cm")), colour="black")+  
 geom\_hline(yintercept = 5, linetype = "dashed", size = 1, color = "#808080") + #med contamination quality = 5-10  
 annotate("text", x=110, y=50, label="High", #angle=270,  
 colour="black", text=element\_text(size=9, family="Arial")) +  
 geom\_segment(aes(x = 105, y = 10, xend = 105, yend = 100),  
 arrow = arrow(length = unit(0.25, "cm")), colour="black")+  
 geom\_hline(yintercept = 10, linetype = "dashed", size = 1, color = "#808080") + #high contamination quality = > 10%  
 xlab("Completeness (%)") +  
 ylab("Contamination (%)") +  
 theme\_Publication\_legend\_side()



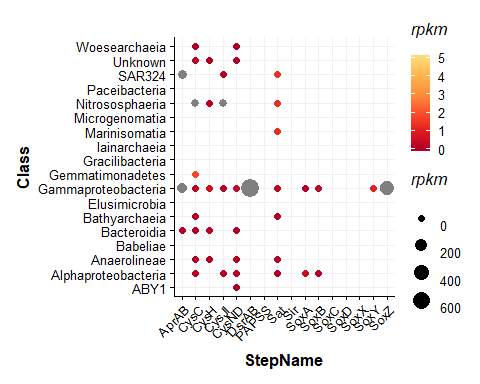
# ----- PLOT WITH ONLY THE DATA WE ARE GOING TO ANALYSE -----  
#This includes the medium-high quality MAGS at depth 200 and the cruises: 48, 72, 73. 74, and 75  
# We are not analyzing Cruise 42, even though we have access to it because all other cruises came from August and cruise 42 came from february. Because the inlet is seasonally anoxic, seasons affect the compostion and we didn't have enough data points to generalize for other seasons than summer (August)  
  
rpkm\_dat %>%  
 dplyr::filter(Contamination <= 10) %>%  
 dplyr::filter(Completeness > 50) %>%  
 ggplot(aes(x=Completeness, y=Contamination, colour=Class)) +  
 geom\_point(aes(size=g\_rpkm)) +  
 #-- MED COMPLETENESS--  
 geom\_vline(xintercept = 50, linetype = "dashed", size = 1, color = "#808080") + #med completeness quality  
 annotate("text", x=70, y=12, label="Medium",   
 colour="black", text=element\_text(size=9, family="Arial")) +  
 geom\_segment(aes(x = 51, y = 11, xend = 89, yend = 11),  
 arrow = arrow(length = unit(0.25, "cm")), colour="black")+  
 geom\_segment(aes(x = 89, y = 11, xend = 51, yend = 11),  
 arrow = arrow(length = unit(0.25, "cm")), colour="black")+  
 #-- HIGH COMPLETENESS --  
 geom\_vline(xintercept = 90, linetype = "dashed", size = 1, color = "#808080") + #high completeness quality  
 annotate("text", x=95, y=12, label="High",   
 colour="black", text=element\_text(size=9, family="Arial")) +  
 geom\_segment(aes(x = 91, y = 11, xend = 100, yend = 11),  
 arrow = arrow(length = unit(0.25, "cm")), colour="black")+  
 geom\_segment(aes(x = 100, y = 11, xend = 91, yend = 11),  
 arrow = arrow(length = unit(0.25, "cm")), colour="black")+  
 #-- LOW CONTAMINATION --  
 annotate("text", x=104, y=2.5, label="Low", #angle=270,  
 colour="black", text=element\_text(size=9, family="Arial")) +  
 geom\_segment(aes(x = 101, y = 0, xend = 101, yend = 4.8),  
 arrow = arrow(length = unit(0.25, "cm")), colour="black")+  
 geom\_segment(aes(x = 101, y = 4.8, xend = 101, yend = 0),  
 arrow = arrow(length = unit(0.25, "cm")), colour="black")+  
 #-- MED CONTAMINATION --  
 annotate("text", x=104, y=8, label="Med", #angle=270,  
 colour="black", text=element\_text(size=9, family="Arial")) +  
 geom\_segment(aes(x = 101, y = 5.2, xend = 101, yend = 10),  
 arrow = arrow(length = unit(0.25, "cm")), colour="black")+  
 geom\_segment(aes(x = 101, y = 10, xend = 101, yend = 5.2),  
 arrow = arrow(length = unit(0.25, "cm")), colour="black")+  
 geom\_hline(yintercept = 5, linetype = "dashed", size = 1, color = "#808080") + #med contamination quality = 5-10  
 xlab("Completeness (%)") +  
 ylab("Contamination (%)") +  
 theme\_Publication\_legend\_side()



## Bubble Plot of Classes

Next I will create RPKM bubble-plot of each Sulphur-cycling gene versus taxonomy

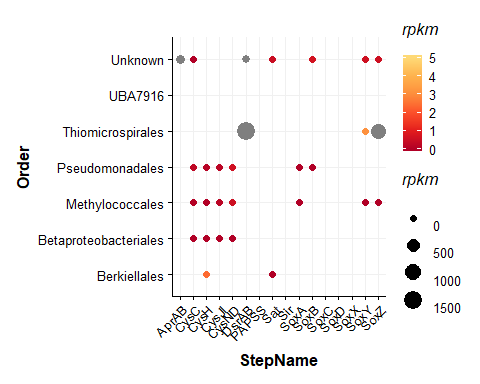
#Data frame of ko values and their corresponding step in the pathway (found manually from pathview output)  
ko\_to\_step <- read\_csv("KO-StepName-Mapping.csv")  
  
#read in the pathview output from sulfur pathway grouped by class (generated in pathview\_by\_class Rmd file)  
pv\_gene\_dat <- readRDS("pv\_out\_mean.rds")  
  
#Filter by the ko values that were found to be involved in the pathways steps we are interested in  
pv\_steps\_dat <- dplyr::inner\_join(pv\_gene\_dat, ko\_to\_step)  
pv\_steps\_dat <- pv\_steps\_dat[, -grep(".col$", colnames(pv\_steps\_dat))]  
names(pv\_steps\_dat)[names(pv\_steps\_dat) == 'c\_\_'] <- 'c\_\_Unknown'  
  
#wide to long format  
pv\_steps\_dat <- pv\_steps\_dat %>%  
 tidyr::gather(key = Class, value = rpkm, -c(kegg.names, labels, KO, all.mapped, type, width, height, StepName, x, y))  
  
#Create a bubble plot  
pv\_steps\_dat %>%  
 mutate(Class = gsub("c\_\_", "", Class)) %>%  
 ggplot(aes(x=StepName, y=Class, size = rpkm, color = rpkm)) +   
 geom\_point() +   
 theme(axis.text.x = element\_text(angle = 90, hjust = 1)) +  
 scale\_color\_gradientn(limits = c(0,5),  
 colours = c("#b10026", "#e31a1c","#fc4e2a", "#fd8d3c", "#feb24c", "#fed976"),  
 breaks = c(0,1,2,3,4,5),  
 guide = guide\_colorbar(barwidth = 1, barheight = 5)) +  
 theme\_Publication\_legend\_side() +  
 scale\_size(range=c(2, 6)) +  
 theme(axis.text.x = element\_text(angle = 45, hjust = 1),  
 legend.text = element\_text(margin = margin(t = 10)),  
 legend.spacing.y = unit(0.5, 'cm'))



## Bubble Plot from gamma proteobacteria

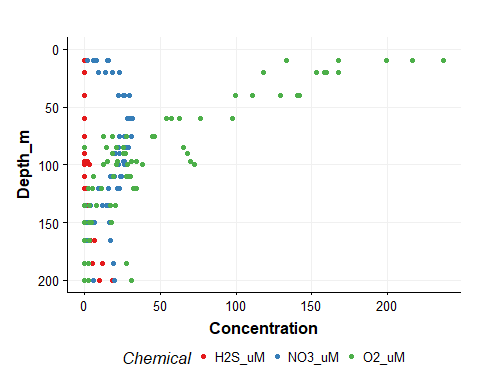
Next I will create RPKM bubble-plot of each Nitrogen/Sulphur-cycling gene versus taxonomy

#Data frame of ko values and their corresponding step in the pathway (found manually from pathview output)  
ko\_to\_step <- read\_csv("KO-StepName-Mapping.csv")  
  
#read in the pathview output from sulfur pathway grouped by class (generated in pathview\_by\_class Rmd file)  
pv\_gene\_dat <- readRDS("gamma\_pv\_out.rds")  
  
#Filter by the ko values that were found to be involved in the pathways steps we are interested in  
pv\_steps\_dat <- dplyr::inner\_join(pv\_gene\_dat, ko\_to\_step)  
pv\_steps\_dat <- pv\_steps\_dat[, -grep(".col$", colnames(pv\_steps\_dat))]  
names(pv\_steps\_dat)[names(pv\_steps\_dat) == 'o\_\_'] <- 'o\_\_Unknown'  
  
#wide to long format  
pv\_steps\_dat <- pv\_steps\_dat %>%  
 tidyr::gather(key = Order, value = rpkm, -c(kegg.names, labels, KO, all.mapped, type, width, height, StepName, x, y))  
  
#Create a bubble plot  
pv\_steps\_dat %>%  
 mutate(Order = gsub("o\_\_", "", Order)) %>%  
 ggplot(aes(x=StepName, y=Order)) +   
 geom\_point(aes(size = rpkm, color=rpkm)) +   
 theme(axis.text.x = element\_text(angle = 90, hjust = 1)) +  
 scale\_color\_gradientn(limits = c(0,5),  
 colours = c("#b10026", "#e31a1c","#fc4e2a", "#fd8d3c", "#feb24c", "#fed976"),  
 breaks = c(0,1,2,3,4,5),  
 guide = guide\_colorbar(barwidth = 1, barheight = 5)) +  
 theme\_Publication\_legend\_side() +  
 scale\_size(range=c(2, 6)) +  
 theme(axis.text.x = element\_text(angle = 45, hjust = 1),  
 legend.text = element\_text(margin = margin(t = 10)),  
 legend.spacing.y = unit(0.5, 'cm'))

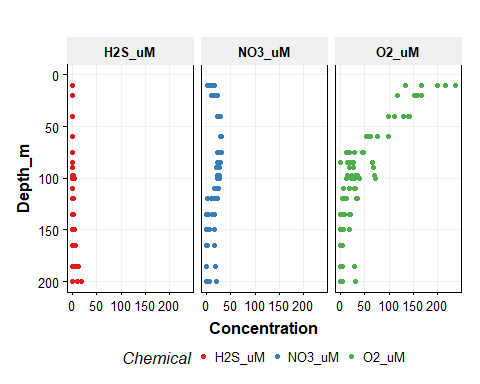


## Geochemical Gradients

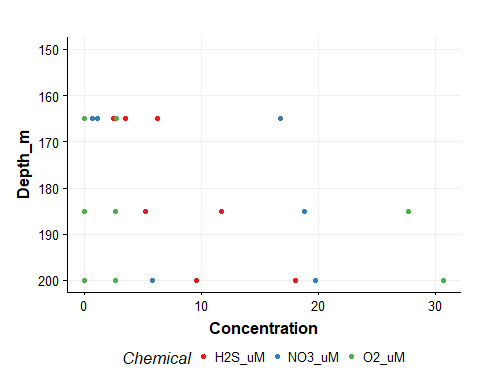
# Load raw data  
raw\_dat <- readr::read\_csv("Saanich\_Data.csv")  
  
#Clean dat  
dat <-   
 raw\_dat %>%  
 dplyr::select(Cruise, Date, Depth, Temperature,  
 WS\_O2, WS\_NO3, WS\_H2S) %>%  
 dplyr::filter(!is.na(WS\_O2)) %>%  
 dplyr::rename(O2\_uM=WS\_O2, NO3\_uM=WS\_NO3, H2S\_uM=WS\_H2S) %>%  
 dplyr::mutate(Depth\_m=Depth\*1000) %>%  
 dplyr::filter(Cruise %in% c(48, 72, 73, 74, 75)) #Only take from the relevant cruises  
  
###############  
# -- First let's look at all of the depths, so we can get a high level picture of the concentration changes with depth --  
#want to create a plot with all 3 variables as x axis and depth as y with shape as type  
#can manipulate data frame to do so  
dat %>%  
 dplyr::select(Depth\_m, H2S\_uM, NO3\_uM, O2\_uM) %>%  
 gather(key = "Chemical", value = "Concentration", -Depth\_m) %>%  
 ggplot() +  
 geom\_point(aes(x=Concentration, y=Depth\_m, colour=Chemical)) +  
 scale\_y\_reverse() +  
 expand\_limits(y=0) +  
 theme\_Publication() +  
 scale\_color\_manual(values=c("#e41a1c", "#377eb8", "#4daf4a"))



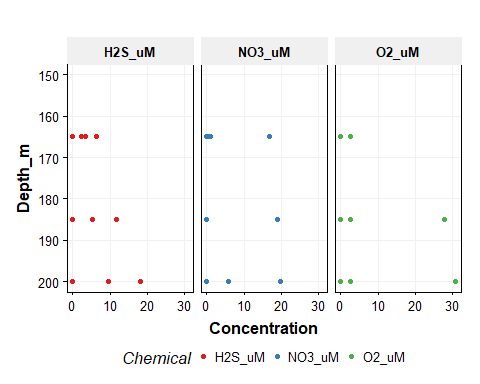
# -- Now let's facet to get a better look --  
dat %>%  
 dplyr::select(Depth\_m, H2S\_uM, NO3\_uM, O2\_uM) %>%  
 gather(key = "Chemical", value = "Concentration", -Depth\_m) %>%  
 ggplot() +  
 geom\_point(aes(x=Concentration, y=Depth\_m, colour=Chemical)) +  
 scale\_y\_reverse() +  
 expand\_limits(y=0) +  
 theme\_Publication\_facets() +  
 scale\_color\_manual(values=c("#e41a1c", "#377eb8", "#4daf4a")) +  
 facet\_wrap(facets = vars(Chemical))



###############  
# -- Next let's zoom into depth 200 --  
  
#Normal  
dat %>%  
 dplyr::select(Depth\_m, H2S\_uM, NO3\_uM, O2\_uM) %>%  
 dplyr::filter(Depth\_m > 150) %>%  
 dplyr::filter(Depth\_m <= 200) %>%  
 gather(key = "Chemical", value = "Concentration", -Depth\_m) %>%  
 ggplot() +  
 geom\_point(aes(x=Concentration, y=Depth\_m, colour=Chemical)) +  
 scale\_y\_reverse() +  
 expand\_limits(y=150) +  
 theme\_Publication() +  
 scale\_color\_manual(values=c("#e41a1c", "#377eb8", "#4daf4a"))



#facet  
dat %>%  
 dplyr::select(Depth\_m, H2S\_uM, NO3\_uM, O2\_uM) %>%  
 dplyr::filter(Depth\_m > 150) %>%  
 dplyr::filter(Depth\_m <= 200) %>%  
 gather(key = "Chemical", value = "Concentration", -Depth\_m) %>%  
 ggplot() +  
 geom\_point(aes(x=Concentration, y=Depth\_m, colour=Chemical)) +  
 scale\_y\_reverse() +  
 expand\_limits(y=150) +  
 theme\_Publication\_facets() +  
 scale\_color\_manual(values=c("#e41a1c", "#377eb8", "#4daf4a")) +  
 facet\_wrap(facets = vars(Chemical))



## Relative Abundance of MAGs across Cruises

rpkm\_dat <- left\_join(metag\_rpkm, checkm\_dat, by="mag") %>%   
 left\_join(gtdb\_dat, by="mag") %>%   
 filter(Completeness > 50 & Contamination < 10) %>% # good quality MAGs  
 mutate(mag = reorder(mag, Class, sort)) # sort by their taxonomic Order so everything shows up together  
ggplot(rpkm\_dat, aes(x=Sample, y=mag, col=Class)) +  
 geom\_point(aes(size=g\_rpkm)) +  
 theme\_Publication\_legend\_side() +  
 theme(axis.text.x = element\_text(angle = 45,  
 hjust = 1))

