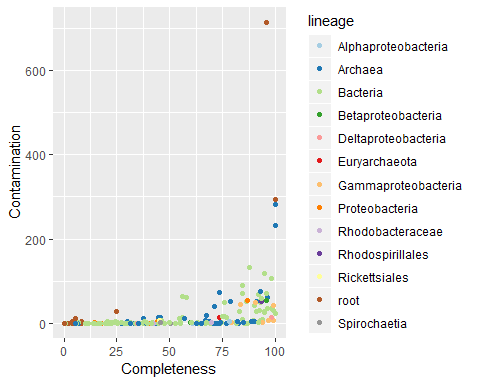
MICB 405 Project 2

Shannah

November 17, 2018

First I will analyse the MAGs from checkM output

#Probably want to include as many MAGs as possible because we are wanting to fill in a pathway  
  
#Load .tsv file from MetaBAT2 to dat  
bin\_dat <- read\_tsv(file="MetaBAT2\_SaanichInlet\_200m\_min1500\_checkM\_stdout.tsv", col\_names = TRUE)  
  
  
#Rename Marker lineage so it's one string (makes it easier later)  
bin\_dat <- bin\_dat %>%  
 dplyr::rename('Marker\_lineage' = 'Marker lineage')  
  
#Add new column with just lineage for point colors  
bin\_dat <- bin\_dat %>%  
 dplyr::mutate('lineage' = gsub(".\*\_\_(\*)", "\\1", Marker\_lineage),  
 'lineage' = gsub("\\(|\\)", "", lineage),  
 'lineage' = gsub("UID.\*", "", lineage))  
  
#Plotting Completeness vs Contamination of MAGs at 200m  
bin\_dat %>%  
 ggplot(aes(x=Completeness, y=Contamination, colour=lineage)) +  
 geom\_point() +  
 scale\_color\_manual(values = c('#a6cee3','#1f78b4',"#b2df8a","#33a02c","#fb9a99",  
 "#e31a1c",'#fdbf6f',"#ff7f00","#cab2d6","#6a3d9a",  
 "#ffff99","#b15928", "#969696")) #added discrete scales ; palette from colorbrewer2 but added extra ("#969696") b/c 13 values



Visualizing the metabolic map with pathview This tutorial is meant to take your KEGG Ortholog (KO) annotation tables from KAAS, and one of the rpkm files (.csv format) and view a pathway or metabolic map with these data layered on.

Before you begin loading, transforming, and playing with data we will need to load the following libraries at the very least:

Now we are going to read our tabular files. First is KO annotation tables (query.ko.txt) then the rpkm file. I’m using rename to assign names to each of the columns as these do not have headers.

ko <- read.table("SaanichInlet\_MAGs\_ORFs\_ko.cleaned.txt") %>%   
 dplyr::rename(orf = V1) %>%   
 dplyr::rename(ko = V2)  
  
#TODO: make a loop?  
rpkm\_42 <- read.csv("SI042\_200m.RPKM.csv", header=FALSE) %>%   
 dplyr::rename(orf = V1) %>%   
 dplyr::rename(rpkm = V2)  
rpkm\_48 <- read.csv("SI048\_200m.RPKM.csv", header=FALSE) %>%   
 dplyr::rename(orf = V1) %>%   
 dplyr::rename(rpkm = V2)  
rpkm\_73 <- read.csv("SI073\_200m.RPKM.csv", header=FALSE) %>%   
 dplyr::rename(orf = V1) %>%   
 dplyr::rename(rpkm = V2)  
rpkm\_74 <- read.csv("SI074\_200m.RPKM.csv", header=FALSE) %>%   
 dplyr::rename(orf = V1) %>%   
 dplyr::rename(rpkm = V2)  
rpkm\_75 <- read.csv("SI075\_200m.RPKM.csv", header=FALSE) %>%   
 dplyr::rename(orf = V1) %>%   
 dplyr::rename(rpkm = V2)

Here are the tables in their current form, ready for merging by orf:

## orf ko  
## 1 EBFFFBAL\_00003 K07478  
## 2 EBFFFBAL\_00004 K00278  
## 3 EBFFFBAL\_00005 K03517  
## 4 EBFFFBAL\_00006 K00767  
## 5 EBFFFBAL\_00008 K07090  
## 6 EBFFFBAL\_00010 K07025

## orf rpkm  
## 1 AALJDOEL\_00001 0  
## 2 AALJDOEL\_00002 0  
## 3 AALJDOEL\_00003 0  
## 4 AALJDOEL\_00004 0  
## 5 AALJDOEL\_00005 0  
## 6 AALJDOEL\_00006 0

## orf rpkm  
## 1 AALJDOEL\_00001 0  
## 2 AALJDOEL\_00002 0  
## 3 AALJDOEL\_00003 0  
## 4 AALJDOEL\_00004 0  
## 5 AALJDOEL\_00005 0  
## 6 AALJDOEL\_00006 0

## orf rpkm  
## 1 AALJDOEL\_00001 0  
## 2 AALJDOEL\_00002 0  
## 3 AALJDOEL\_00003 0  
## 4 AALJDOEL\_00004 0  
## 5 AALJDOEL\_00005 0  
## 6 AALJDOEL\_00006 0

## orf rpkm  
## 1 AALJDOEL\_00001 0  
## 2 AALJDOEL\_00002 0  
## 3 AALJDOEL\_00003 0  
## 4 AALJDOEL\_00004 0  
## 5 AALJDOEL\_00005 0  
## 6 AALJDOEL\_00006 0

## orf rpkm  
## 1 AALJDOEL\_00001 0.00000  
## 2 AALJDOEL\_00002 0.00000  
## 3 AALJDOEL\_00003 0.00000  
## 4 AALJDOEL\_00004 2.21173  
## 5 AALJDOEL\_00005 0.00000  
## 6 AALJDOEL\_00006 0.00000

Next steps are to join these two tables and some basic transformations to make things a bit easier for pathview. I’ve separated, or split, the orf value of each row into two new variables: mag and orf\_id corresponding to the character string before and after the underscore in orf. This makes it easier to group\_by MAGs and will be necessary for joining other tables (such as checkM, gtdbtk, etc.) into one dataframe (to rule them all).

The summarise code is summing all RPKM values assigned to a KO number for each MAG. This is useful to prevent multiple rows in an eventual matrix for pathview for each copy found. Or accidentally dropping those data if we’re not careful. Anyway, we can freely sum RPKM values and that is what is easiest here.

**NOTE**: If your are dealing with RPKM values from multiple cruises (in this example I am only dealing with RPKM from SI042) you will also need to group by a cruise variable so these are not summed. Or maybe you want them to be if you are not interested in the time/season/cruise variable. If you are interested in visualizing the variability in transcription of a single MAG across the cruises you may also want to filter for your MAG of interest then group by ko and cruise. It all depends on what question you want to answer so be mindful here!

#rpkm\_42, rpkm\_48, rpkm\_73, rpkm\_74, rpkm\_75  
  
ko\_rpkm\_42 <- left\_join(ko, rpkm\_42, by="orf") %>%   
 separate(orf, into=c("mag", "orf\_id")) # Split the Prokka ORF names into MAG identifier and ORF  
ko\_rpkm\_48 <- left\_join(ko, rpkm\_48, by="orf") %>%   
 separate(orf, into=c("mag", "orf\_id")) # Split the Prokka ORF names into MAG identifier and ORF  
ko\_rpkm\_73 <- left\_join(ko, rpkm\_73, by="orf") %>%   
 separate(orf, into=c("mag", "orf\_id")) # Split the Prokka ORF names into MAG identifier and ORF  
ko\_rpkm\_74 <- left\_join(ko, rpkm\_74, by="orf") %>%   
 separate(orf, into=c("mag", "orf\_id")) # Split the Prokka ORF names into MAG identifier and ORF  
ko\_rpkm\_75 <- left\_join(ko, rpkm\_75, by="orf") %>%   
 separate(orf, into=c("mag", "orf\_id")) # Split the Prokka ORF names into MAG identifier and ORF  
#Merge all  
ko\_rpkm <- rbind(ko\_rpkm\_42, ko\_rpkm\_48, ko\_rpkm\_73, ko\_rpkm\_74, ko\_rpkm\_75)  
  
#number for joining  
t\_rpkm <- ko\_rpkm %>%   
 group\_by(mag, ko) %>%   
 summarise(total = sum(rpkm))  
rpkm\_mat <- spread(t\_rpkm, key = mag, value = total)

As an example, we can view the nitrogen metabolism capabilities of our MAGs. To view a different pathway or metabolism the pathway.id parameter will need to be changed. Searching for your pathway of interest via the KEGG browser is likely the easiest way to find these IDs.

We can also view the dataframe that is generated by pathview. Unfortunately it is not that interesting or useful.

# Nitrogen metabolism  
n.pv <- pathview(gene.data = rpkm\_mat,  
 species = "ko",  
 pathway.id="00910",  
 kegg.dir = "C:\\Users\\Shannah\\OneDrive\\Uni-Year-5\\MICB\_405\\Project2")  
head(n.pv$plot.data.gene)

## kegg.names labels all.mapped type x y width height ko mol.col  
## 10 K01455 K01455 ortholog 641 129 46 17 NA #FFFFFF  
## 22 K10535 K10535 ortholog 417 357 46 17 NA #FFFFFF  
## 23 K10534 K10534 ortholog 225 287 46 17 NA #FFFFFF  
## 25 K00360 K00370 ortholog 248 316 46 17 NA #FFFFFF  
## 26 K00367 K00367 ortholog 225 266 46 17 NA #FFFFFF  
## 27 K00459 K00459 ortholog 307 376 46 17 NA #FFFFFF

Here is a file that pathview automatically writes to your kegg.dir directory.

“An example nitrogen metabolism wiring diagram derived from Saanich Inlet MAGs. There are no negative RPKM values and therefore the lowest value is 0. Vertical red bars indicate some MAGs are actively transcirbing genes involved in disimilatory nitrate reduction, denitrification and nitrification.”

“An example nitrogen metabolism wiring diagram derived from Saanich Inlet MAGs. There are no negative RPKM values and therefore the lowest value is 0. Vertical red bars indicate some MAGs are actively transcirbing genes involved in disimilatory nitrate reduction, denitrification and nitrification.”

```