Formatieve opdracht

Thijmen Weijgertze

4/3/2023

# Species identification and abundance estimation

### FastQC analysis

# update conda and answer y to proceed question  
echo “y” | conda update -n base conda

# show all conda virtual environments  
conda env list  
  
# show yml file content  
cat ~/daur2/metagenomics/yml/setup\_meta\_env.yml

## bash: cannot set terminal process group (569277): Inappropriate ioctl for device  
## bash: no job control in this shell  
## # conda environments:  
## #  
## base \* /home/thijmen.weijgertze/miniconda3  
## kraken\_biom\_v101 /home/thijmen.weijgertze/miniconda3/envs/kraken\_biom\_v101  
## meta /home/thijmen.weijgertze/miniconda3/envs/meta  
## meta2 /home/thijmen.weijgertze/miniconda3/envs/meta2  
##   
## name: meta  
## dependencies:  
## - python=3.8

# create conda virtual environment  
conda env create --file ~/daur2/metagenomics/yml/setup\_meta\_env.yml

# activate environment  
conda activate meta  
  
# install fastqc and answer yes for procceed question  
echo "y" | conda install -c bioconda fastqc  
  
# deactivate conda virtual environment  
conda deactivate

# activate environment  
conda activate meta  
  
# perform FastQC analysis  
fastqc -o ~/daur2/metagenomics/formatieve\_opdracht/fastqc/ /home/daur2/metagenomics/formative\_data/HU2\_MOCK2\_L001\_R1\_001.fastq.gz  
fastqc -o ~/daur2/metagenomics/formatieve\_opdracht/fastqc/ /home/daur2/metagenomics/formative\_data/HU2\_MOCK2\_L001\_R2\_001.fastq.gz  
  
# deactivate conda virtual environment  
conda deactivate

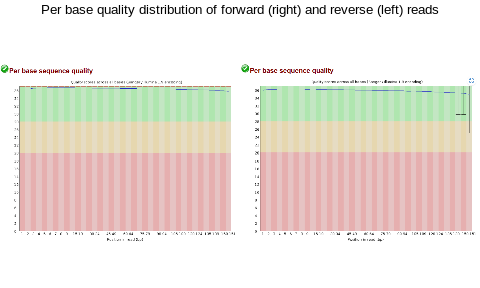


Fig. 1: Quality distribution: Left-side) Foward read quality scores range from 0 to 40 on the Phred scale. All bases for all reads have very high quality (>30), indicating that our reads have an accuracy of >99,9%. Right-side) reversed read quality scores range from 0 to 40 on the Phred scale. All bases for all reads have very high quality (>30). This indicates our reads have an accuracy of >99,9%

### Species identification using minikraken

# activate environment  
conda activate meta  
  
#install kraken2 and answer y to proceed question  
echo "y" | conda install -c bioconda kraken2  
  
conda deactivate

# activate environment  
conda activate meta  
  
kraken2 --db /home/daur2/metagenomics/minikraken2\_v2\_8GB\_201904\_UPDATE/ --threads 2 --paired --gzip-compressed --output ~/daur2/metagenomics/formatieve\_opdracht/kraken2/mock2.kraken --report ~/daur2/metagenomics/formatieve\_opdracht/kraken2/mock2.report --use-names /home/daur2/metagenomics/formative\_data/HU2\_MOCK2\_L001\_R1\_001.fastq.gz /home/daur2/metagenomics/formative\_data/HU2\_MOCK2\_L001\_R2\_001.fastq.gz  
  
conda deactivate

Kraken2 identification statistics:

* Classified: 59541308 sequences classified (90.33%)
* Unclassified: 6374528 sequences unclassified (9.67%)
* Threads: 1
* Duration: 65915836 sequences (19439.90 Mbp) processed in 1734.357s (2280.4 Kseq/m, 672.52 Mbp/m)
* .kraken size: 13.3 GB
* .report size: 272 KB

# activate environment  
conda activate meta  
  
# installing bioconda bracken  
echo "y" | conda install -c bioconda bracken  
  
conda deactivate

# activate environment  
conda activate meta  
  
# bracken  
bracken -d /home/daur2/metagenomics/minikraken2\_v2\_8GB\_201904\_UPDATE/ -i ~/daur2/metagenomics/formatieve\_opdracht/kraken2/mock2.report -o ~/daur2/metagenomics/formatieve\_opdracht/bracken/mock2.bracken  
  
conda deactivate

# activate environment  
conda activate kraken\_biom\_v101  
  
# install kraken-biom and answer y to proceed question  
echo "y" | conda install -c "bioconda/label/cf201901" kraken-biom  
  
conda deactivate

# activate environment  
conda activate kraken\_biom\_v101  
  
kraken-biom ~/daur2/metagenomics/formatieve\_opdracht/kraken2/mock2\_bracken\_species.report --fmt json -o ~/daur2/metagenomics/formatieve\_opdracht/kraken2/mock2\_bracken\_species.biom  
  
conda deactivate

Number of species in sample: 1705

### visualisation of data

# load nessecary packages   
BiocManager::install("phyloseq")

library(phyloseq)  
library(ggplot2)  
data <- "~/daur2/metagenomics/formatieve\_opdracht/kraken2/mock2\_bracken\_species.biom"  
  
merged\_metagenomes <- import\_biom(data)  
#View(merged\_metagenomes@tax\_table@.Data)

# remove taxonomy information from values/ remove first four characters  
merged\_metagenomes@tax\_table@.Data <-  
 substring(merged\_metagenomes@tax\_table@.Data, 4)  
  
# rename column header to informative format  
colnames(merged\_metagenomes@tax\_table@.Data) <-   
 c("Kingdom", "Phylum", "Class", "Order", "Family", "Genus", "Species")  
  
# inspect data format  
# View(merged\_metagenomes@tax\_table@.Data)  
  
# using a function subset\_taxa() we can separate the bacterial data and store it in a separate variable:  
merged\_metagenomes\_b <- subset\_taxa(merged\_metagenomes, Kingdom == "Bacteria")  
  
# then using a function called sample\_sums() we can count how many reads are classified as bacterial:  
sample\_sums(merged\_metagenomes\_b)

## sa1   
## 59493788

# to provide informative headers use the function sample\_names():  
sample\_names(merged\_metagenomes\_b) <- "bacteria"

# retrieve the present taxonomic kingdoms from our data  
unique(merged\_metagenomes@tax\_table@.Data[,"Kingdom"])

## [1] "Bacteria" "Eukaryota" "Archaea" "Viruses"

# subset kingdoms  
merged\_metagenomes\_b <- subset\_taxa(merged\_metagenomes, Kingdom == "Bacteria")  
merged\_metagenomes\_e <- subset\_taxa(merged\_metagenomes, Kingdom == "Eukaryota")  
merged\_metagenomes\_a <- subset\_taxa(merged\_metagenomes, Kingdom == "Archaea")  
merged\_metagenomes\_v <- subset\_taxa(merged\_metagenomes, Kingdom == "Viruses")  
  
# rename samples  
sample\_names(merged\_metagenomes\_b) <- "bacteria"  
sample\_names(merged\_metagenomes\_e) <- "eukaryote"  
sample\_names(merged\_metagenomes\_a) <- "archea"  
sample\_names(merged\_metagenomes\_v) <- "virus"  
  
# count number of reads in total  
sample\_sums(merged\_metagenomes)

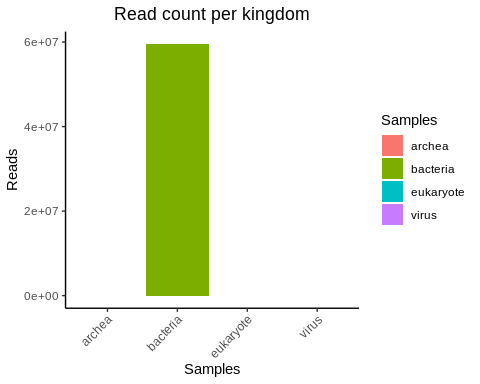
## sa1   
## 59540195

# count number of reads per kingdom  
c(sample\_sums(merged\_metagenomes\_b), sample\_sums(merged\_metagenomes\_e), sample\_sums(merged\_metagenomes\_a), sample\_sums(merged\_metagenomes\_v))

## bacteria eukaryote archea virus   
## 59493788 18373 280 27754

# collect kingdoms seperately in data frames  
data\_b <- data.frame(Samples = sample\_names(merged\_metagenomes\_b),  
 Reads = sample\_sums(merged\_metagenomes\_b))  
  
data\_e <- data.frame(Samples = sample\_names(merged\_metagenomes\_e),  
 Reads = sample\_sums(merged\_metagenomes\_e))  
  
data\_a <- data.frame(Samples = sample\_names(merged\_metagenomes\_a),  
 Reads = sample\_sums(merged\_metagenomes\_a))  
  
data\_v <- data.frame(Samples = sample\_names(merged\_metagenomes\_v),  
 Reads = sample\_sums(merged\_metagenomes\_v))  
  
# merge kingdom data in to one data frame  
data\_t <- rbind(data\_b, data\_e, data\_a, data\_v)

# plot number of reads per kingdom  
ggplot(data = data\_t, mapping = aes(x = Samples, y = Reads, fill = Samples )) +  
 geom\_col() +  
 theme\_classic() +  
 ggtitle("Read count per kingdom") +  
 theme(plot.title = element\_text(hjust = 0.5)) +  
 theme(axis.text.x=element\_text(angle=45, hjust=1))



# reload the biom data, edit column headers and sample name  
merged\_metagenomes <- import\_biom(data)  
merged\_metagenomes@tax\_table@.Data <- substring(merged\_metagenomes@tax\_table@.Data, 4)  
colnames(merged\_metagenomes@tax\_table@.Data)<- c("Kingdom", "Phylum", "Class", "Order", "Family", "Genus", "Species")  
colnames(merged\_metagenomes@otu\_table) <- c("mock1")  
  
# check empty labels (empty = TRUE)  
summary(merged\_metagenomes@tax\_table@.Data== "")

## Kingdom Phylum Class Order   
## Mode :logical Mode :logical Mode :logical Mode :logical   
## FALSE:1705 FALSE:1675 FALSE:1645 FALSE:1701   
## TRUE :30 TRUE :60 TRUE :4   
## Family Genus Species   
## Mode :logical Mode :logical Mode :logical   
## FALSE:1699 FALSE:1690 FALSE:1705   
## TRUE :6 TRUE :15

# transform data frame for plotting  
glom <- tax\_glom(merged\_metagenomes, taxrank = "Species")  
mock1\_metagenome\_species <- psmelt(glom)  
  
# combine genus and species label for scientific naming in plot  
mock1\_metagenome\_species$Species <- as.character(mock1\_metagenome\_species$Species)  
mock1\_metagenome\_species$Species <- paste(mock1\_metagenome\_species[,]$Genus,mock1\_metagenome\_species[,]$Species, sep=" ", collapse=NULL)  
# check label manipulation  
# unique(mock1\_metagenome\_species$Species)

library(tidyverse)

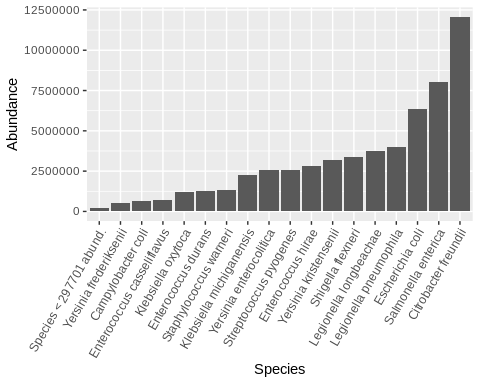
## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.1 ──

## ✓ tibble 3.1.6 ✓ dplyr 1.0.8  
## ✓ tidyr 1.2.0 ✓ stringr 1.4.0  
## ✓ readr 2.1.2 ✓ forcats 0.5.1  
## ✓ purrr 0.3.4

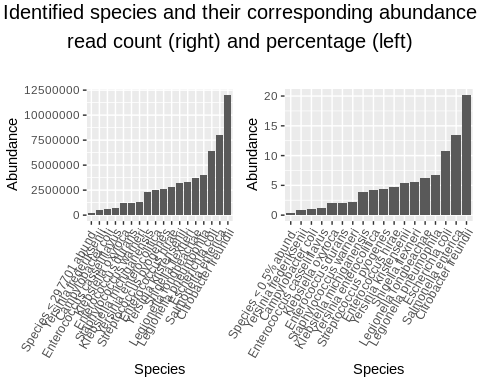
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## x dplyr::combine() masks gridExtra::combine()  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

# alle abundance waarden onder 160.000 opslaan onder één label  
mock1\_metagenome\_species$Species[mock1\_metagenome\_species$Abundance < 297701] <- "Species < 297701 abund."  
  
mock1\_metagenome\_species$Species <- factor(mock1\_metagenome\_species$Species, levels = unique(mock1\_metagenome\_species$Species))

id\_species <- mock1\_metagenome\_species %>%   
 mutate("species\_ordered"=reorder(mock1\_metagenome\_species$Species,mock1\_metagenome\_species$Abundance)) %>%   
 ggplot(aes(x=species\_ordered, y=Abundance))+   
 geom\_col(position = position\_dodge())+  
 theme(axis.text.x = element\_text(angle = 60, hjust = 1), legend.position = "none")+  
 labs(  
 x = "Species"  
 )  
id\_species



# transform read count to percentages  
glom <- tax\_glom(merged\_metagenomes, taxrank = "Species")  
mock1\_metagenome\_species\_percent <- psmelt(glom)  
  
# normalize read counts to percentages  
mock1\_metagenome\_species\_percent$Abundance <- (mock1\_metagenome\_species\_percent$Abundance\*100)/sum(mock1\_metagenome\_species\_percent$Abundance)  
  
# combine genus and species label for scientific naming in plot  
mock1\_metagenome\_species\_percent$Species <- as.character(mock1\_metagenome\_species\_percent$Species)  
mock1\_metagenome\_species\_percent$Species <- paste(mock1\_metagenome\_species\_percent[,]$Genus,mock1\_metagenome\_species\_percent[,]$Species, sep=" ", collapse=NULL)  
mock1\_metagenome\_species\_percent$Species[mock1\_metagenome\_species\_percent$Abundance < 0.5] <- "Species < 0.5% abund."  
#unique(mock1\_metagenome\_species\_percent$Species)  
  
# plotten percent  
id\_species\_percent <- mock1\_metagenome\_species\_percent %>%   
 mutate("species\_ordered"=reorder(mock1\_metagenome\_species\_percent$Species,mock1\_metagenome\_species\_percent$Abundance)) %>%   
   
 ggplot(aes(x=species\_ordered, y=Abundance))+   
 geom\_col(position = position\_dodge())+  
 theme(axis.text.x = element\_text(angle = 60, hjust = 1), legend.position = "none")+  
 labs(  
 x = "Species"  
 )  
  
# id plotten samenvoegen  
grid.arrange(id\_species, id\_species\_percent, ncol=2, top=textGrob("Identified species and their corresponding abundance\nread count (right) and percentage (left)\n", gp=gpar(fontsize=15,font=8)))



# collect composition information  
mock1\_composition <- as.data.frame(read.csv('/home/daur2/metagenomics/reader\_data/HU\_waternet\_MOCK1\_composition.csv', row.names=1, sep = ";"))

mock1\_composition$amount.. <- as.numeric(gsub(",", ".", mock1\_composition$amount..))  
colnames(mock1\_composition) <- c( "name","amount","amountP","sample\_name","total\_volume")

# intersect data bases to collect overlap between composition and `kraken2` results  
mock1\_and\_composition\_intersect <- mock1\_metagenome\_species\_percent[mock1\_metagenome\_species\_percent$Species %in% mock1\_composition$name,]  
  
# collect species that are in composition and not in `kraken2` results  
`%!in%` <- Negate(`%in%`) # allow not in = !in  
comp\_not\_in\_mock1 <- mock1\_composition[mock1\_composition$name %!in% mock1\_metagenome\_species\_percent$Species,]  
  
# generate a list of species overlap between mock1 and composition  
unique(mock1\_and\_composition\_intersect$Species)

## [1] "Citrobacter freundii" "Escherichia coli"   
## [3] "Legionella pneumophila" "Legionella longbeachae"   
## [5] "Shigella flexneri" "Yersinia kristensenii"   
## [7] "Enterococcus hirae" "Streptococcus pyogenes"   
## [9] "Enterococcus durans" "Klebsiella oxytoca"   
## [11] "Enterococcus casseliflavus" "Campylobacter coli"

# generate a list of species that are from composition and not in mock1  
unique(comp\_not\_in\_mock1$name)

## [1] "Campylobacter jejuni" "Klebsiella pneumoniae"   
## [3] "Legionella bozemanii" "Legionella jordanis"   
## [5] "Legionella lansingensis" "Legionella paisiensis"   
## [7] "Salmonella hyphia" "Staphyloccocus epidermidis"

# compare labels between `kraken2` subset and composition and store percent values when labels are identical  
mock1\_and\_composition\_intersect$amountP <- NA  
for (m1\_label in mock1\_and\_composition\_intersect$Species){  
 for (m1c\_label in mock1\_composition$name){  
 if(m1\_label == m1c\_label){  
 mock1\_and\_composition\_intersect$amountP[mock1\_and\_composition\_intersect$Species == m1\_label] <- mock1\_composition$amountP[mock1\_composition$name == m1c\_label]  
 }  
 }  
}

# load library for melting data frame  
library(reshape2)

##   
## Attaching package: 'reshape2'

## The following object is masked from 'package:tidyr':  
##   
## smiths

# collect plotting info and pretify layout format  
#colnames(mock1\_and\_composition\_intersect)  
mock1\_and\_comp\_plotting\_data <- mock1\_and\_composition\_intersect[,c(10,3,11)]  
colnames(mock1\_and\_comp\_plotting\_data) <- c("species", "k\_abundance", "c\_abundance")  
  
mock1\_and\_comp\_plotting\_data$species <- reorder(mock1\_and\_comp\_plotting\_data$species, mock1\_and\_comp\_plotting\_data$k\_abundance)  
  
mock1\_and\_comp\_plotting\_data <- melt(mock1\_and\_comp\_plotting\_data, id.var = "species")  
mock1\_and\_comp\_plotting\_data$value <- as.numeric(mock1\_and\_comp\_plotting\_data$value)  
  
ggplot(mock1\_and\_comp\_plotting\_data, aes(x = species, y = value, fill = variable)) +   
 geom\_bar(aes(), stat="identity", position="dodge") +  
 theme\_classic() +  
 ylab("Abundance (%)") +  
 xlab("") +  
 ggtitle("Abundance comparison between Kraken2 results and composition") +  
 theme(plot.title = element\_text(hjust = 0.5)) +  
 ylim(0,25) +  
 theme(axis.text.x = element\_text(angle = -45, hjust = 0, vjust = 1)) +  
 scale\_fill\_manual(values=c("skyblue", "orangered"))

