

Script pour l'analyse des données du microbiome : des lectures brutes aux analyses communautaires

#Mise à jour de la machine virtuelle

```
sudo apt-get update -y
sudo apt-get install -y libglpk-dev
sudo apt-get install -y liblzma-dev libbz2-dev
```

```
## sudo: unable to resolve host 2409552e9745: Name or service not known
## Hit:1 http://archive.ubuntu.com/ubuntu focal InRelease
## Hit:2 http://security.ubuntu.com/ubuntu focal-security InRelease
## Hit:3 http://archive.ubuntu.com/ubuntu focal-updates InRelease
## Hit:4 http://archive.ubuntu.com/ubuntu focal-backports InRelease
## Reading package lists...
## sudo: unable to resolve host 2409552e9745: Name or service not known
## Reading package lists...
## Building dependency tree...
## Reading state information...
## libglpk-dev is already the newest version (4.65-2).
## 0 upgraded, 0 newly installed, 0 to remove and 11 not upgraded.
## sudo: unable to resolve host 2409552e9745: Name or service not known
## Reading package lists...
## Building dependency tree...
## Reading state information...
## libbz2-dev is already the newest version (1.0.8-2).
## liblzma-dev is already the newest version (5.2.4-1ubuntu1).
## 0 upgraded, 0 newly installed, 0 to remove and 11 not upgraded.
```

#Installation des différents packages

```
if (!requireNamespace("BiocManager", quietly = TRUE))
  install.packages("BiocManager")
BiocManager::install("BiocStyle")
```

```
## 'getOption("repos")' replaces Bioconductor standard repositories, see
## '?repositories' for details
##
## replacement repositories:
##   CRAN: https://packagemanager.rstudio.com/cran/__linux__/focal/latest
```

```
## Bioconductor version 3.14 (BiocManager 1.30.16), R 4.1.2 (2021-11-01)
```

```
## Warning: package(s) not installed when version(s) same as current; use 'force = TRUE' to
## re-install: 'BiocStyle'
```

```
## Installation paths not writeable, unable to update packages
## path: /usr/local/lib/R/library
## packages:
## Matrix
```

```
BiocManager::install("Rhtslib")
```

```
## 'getOption("repos")' replaces Bioconductor standard repositories, see
## '?repositories' for details
##
## replacement repositories:
## CRAN: https://packagemanager.rstudio.com/cran/__linux__/focal/latest
##
## Bioconductor version 3.14 (BiocManager 1.30.16), R 4.1.2 (2021-11-01)
```

```
## Warning: package(s) not installed when version(s) same as current; use 'force = TRUE' to
## re-install: 'Rhtslib'
```

```
## Installation paths not writeable, unable to update packages
## path: /usr/local/lib/R/library
## packages:
## Matrix
```

```
library("knitr")
library("BiocStyle")
.cran_packages <- c("ggplot2", "gridExtra", "devtools")
install.packages(.cran_packages)
```

```
## Installing packages into '/usr/local/lib/R/site-library'
## (as 'lib' is unspecified)
```

```
.bioc_packages <- c("dada2", "DECIPHER", "phangorn", "phyloseq")
BiocManager::install(.bioc_packages)
```

```
## 'getOption("repos")' replaces Bioconductor standard repositories, see
## '?repositories' for details
##
## replacement repositories:
## CRAN: https://packagemanager.rstudio.com/cran/__linux__/focal/latest
##
## Bioconductor version 3.14 (BiocManager 1.30.16), R 4.1.2 (2021-11-01)
```

```
## Warning: package(s) not installed when version(s) same as current; use 'force = TRUE' to
## re-install: 'dada2' 'DECIPHER' 'phangorn' 'phyloseq'
```

```
## Installation paths not writeable, unable to update packages
## path: /usr/local/lib/R/library
## packages:
## Matrix
```

```
supply(c(.cran_packages, .bioc_packages), require, character.only = TRUE)
```

```
## Loading required package: ggplot2
```

```
## Loading required package: gridExtra
```

```
## Loading required package: devtools
```

```
## Loading required package: usethis
```

```
## Loading required package: dada2
```

```
## Loading required package: Rcpp
```

```
## Loading required package: DECIPHER
```

```
## Loading required package: Biostrings
```

```
## Loading required package: BiocGenerics
```

```
##
```

```
## Attaching package: 'BiocGenerics'
```

```
## The following object is masked from 'package:gridExtra':
```

```
##
```

```
##      combine
```

```
## The following objects are masked from 'package:stats':
```

```
##
```

```
##      IQR, mad, sd, var, xtabs
```

```
## The following objects are masked from 'package:base':
```

```
##
```

```
##      anyDuplicated, append, as.data.frame, basename, cbind, colnames,  
##      dirname, do.call, duplicated, eval, evalq, Filter, Find, get, grep,  
##      grepl, intersect, is.unsorted, lapply, Map, mapply, match, mget,  
##      order, paste, pmax, pmax.int, pmin, pmin.int, Position, rank,  
##      rbind, Reduce, rownames, sapply, setdiff, sort, table, tapply,  
##      union, unique, unsplit, which.max, which.min
```

```
## Loading required package: S4Vectors
```

```
## Loading required package: stats4
```

```
##
```

```
## Attaching package: 'S4Vectors'
```

```
## The following objects are masked from 'package:base':
```

```
##
```

```
##      expand.grid, I, unname
```

```

## Loading required package: IRanges

## Loading required package: XVector

## Loading required package: GenomeInfoDb

##
## Attaching package: 'Biostrings'

## The following object is masked from 'package:base':
##
##      strsplit

## Loading required package: RSQLite

## Loading required package: parallel

## Loading required package: phangorn

## Loading required package: ape

##
## Attaching package: 'ape'

## The following object is masked from 'package:Biostrings':
##
##      complement

## Loading required package: phyloseq

##
## Attaching package: 'phyloseq'

## The following object is masked from 'package:IRanges':
##
##      distance

##      ggplot2 gridExtra devtools      dada2 DECIPHER phangorn phyloseq
##      TRUE      TRUE      TRUE      TRUE      TRUE      TRUE      TRUE

library("knitr")
library("BiocStyle")
.cran_packages <- c("ggplot2", "gridExtra")
.bioc_packages <- c("dada2", "phyloseq", "DECIPHER", "phangorn")
# Load packages into session, and print package version
sapply(c(.cran_packages, .bioc_packages), require, character.only = TRUE)

##      ggplot2 gridExtra      dada2 phyloseq DECIPHER phangorn
##      TRUE      TRUE      TRUE      TRUE      TRUE      TRUE

##Lecture brute des données en tableau

```

```
set.seed(100)
miseq_path <- "/home/rstudio/MiSeq_SOP"
list.files(miseq_path)
```

```
## [1] "F3D0_S188_L001_R1_001.fastq" "F3D0_S188_L001_R2_001.fastq"
## [3] "F3D1_S189_L001_R1_001.fastq" "F3D1_S189_L001_R2_001.fastq"
## [5] "F3D141_S207_L001_R1_001.fastq" "F3D141_S207_L001_R2_001.fastq"
## [7] "F3D142_S208_L001_R1_001.fastq" "F3D142_S208_L001_R2_001.fastq"
## [9] "F3D143_S209_L001_R1_001.fastq" "F3D143_S209_L001_R2_001.fastq"
## [11] "F3D144_S210_L001_R1_001.fastq" "F3D144_S210_L001_R2_001.fastq"
## [13] "F3D145_S211_L001_R1_001.fastq" "F3D145_S211_L001_R2_001.fastq"
## [15] "F3D146_S212_L001_R1_001.fastq" "F3D146_S212_L001_R2_001.fastq"
## [17] "F3D147_S213_L001_R1_001.fastq" "F3D147_S213_L001_R2_001.fastq"
## [19] "F3D148_S214_L001_R1_001.fastq" "F3D148_S214_L001_R2_001.fastq"
## [21] "F3D149_S215_L001_R1_001.fastq" "F3D149_S215_L001_R2_001.fastq"
## [23] "F3D150_S216_L001_R1_001.fastq" "F3D150_S216_L001_R2_001.fastq"
## [25] "F3D2_S190_L001_R1_001.fastq" "F3D2_S190_L001_R2_001.fastq"
## [27] "F3D3_S191_L001_R1_001.fastq" "F3D3_S191_L001_R2_001.fastq"
## [29] "F3D5_S193_L001_R1_001.fastq" "F3D5_S193_L001_R2_001.fastq"
## [31] "F3D6_S194_L001_R1_001.fastq" "F3D6_S194_L001_R2_001.fastq"
## [33] "F3D7_S195_L001_R1_001.fastq" "F3D7_S195_L001_R2_001.fastq"
## [35] "F3D8_S196_L001_R1_001.fastq" "F3D8_S196_L001_R2_001.fastq"
## [37] "F3D9_S197_L001_R1_001.fastq" "F3D9_S197_L001_R2_001.fastq"
## [39] "filtered" "HMP MOCK.v35.fasta"
## [41] "Mock_S280_L001_R1_001.fastq" "Mock_S280_L001_R2_001.fastq"
## [43] "mouse.dpw.metadata" "mouse.time.design"
## [45] "stability.batch" "stability.files"
```

#Découpage des données, filtration

```
fnFs <- sort(list.files(miseq_path, pattern="_R1_001.fastq"))
fnRs <- sort(list.files(miseq_path, pattern="_R2_001.fastq"))
```

```
sampleNames <- sapply(strsplit(fnFs, "_"), `[`, 1)
fnFs <- file.path(miseq_path, fnFs)
fnRs <- file.path(miseq_path, fnRs)
fnFs[1:3]
```

```
## [1] "/home/rstudio/MiSeq_SOP/F3D0_S188_L001_R1_001.fastq"
## [2] "/home/rstudio/MiSeq_SOP/F3D1_S189_L001_R1_001.fastq"
## [3] "/home/rstudio/MiSeq_SOP/F3D141_S207_L001_R1_001.fastq"
```

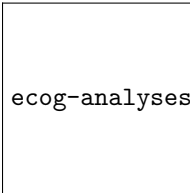
```
plotQualityProfile(fnFs[1:2])
```

```
## Warning: 'guides(<scale> = FALSE)' is deprecated. Please use 'guides(<scale> =
## "none")' instead.
```

ecog-analyses_files/figure-latex/unnamed-chunk-8-1.pdf

```
plotQualityProfile(fnRs[1:2])
```

```
## Warning: 'guides(<scale> = FALSE)' is deprecated. Please use 'guides(<scale> =  
## "none")' instead.
```



ecog-analyses_files/figure-latex/unnamed-chunk-9-1.pdf

```
filt_path <- file.path(miseq_path, "filtered")  
if(!file_test("-d", filt_path)) dir.create(filt_path)  
filtFs <- file.path(filt_path, paste0(sampleNames, "_F_filt.fastq.gz"))  
filtRs <- file.path(filt_path, paste0(sampleNames, "_R_filt.fastq.gz"))
```

```
out <- filterAndTrim(fnFs, filtFs, fnRs, filtRs, truncLen=c(240,160),  
                    maxN=0, maxEE=c(2,2), truncQ=2, rm.phix=TRUE,  
                    compress=TRUE, multithread=TRUE)  
head(out)
```

```
##                               reads.in reads.out  
## F3D0_S188_L001_R1_001.fastq      7793      7113  
## F3D1_S189_L001_R1_001.fastq      5869      5299  
## F3D141_S207_L001_R1_001.fastq     5958      5463  
## F3D142_S208_L001_R1_001.fastq     3183      2914  
## F3D143_S209_L001_R1_001.fastq     3178      2941  
## F3D144_S210_L001_R1_001.fastq     4827      4312
```

La fonction “filterAndTrim” de DADA2 permet de filtrer les données

##Dereplication

```
derepFs <- derepFastq(filtFs, verbose=TRUE)
```

```
## Dereplicating sequence entries in Fastq file: /home/rstudio/MiSeq_SOP/filtered/F3D0_F_filt.fastq.gz
```

```
## Encountered 1979 unique sequences from 7113 total sequences read.
```

```
## Dereplicating sequence entries in Fastq file: /home/rstudio/MiSeq_SOP/filtered/F3D1_F_filt.fastq.gz
```

```
## Encountered 1639 unique sequences from 5299 total sequences read.
```

```
## Dereplicating sequence entries in Fastq file: /home/rstudio/MiSeq_SOP/filtered/F3D141_F_filt.fastq.g
```

```
## Encountered 1477 unique sequences from 5463 total sequences read.
```

```
## Dereplicating sequence entries in Fastq file: /home/rstudio/MiSeq_SOP/filtered/F3D142_F_filt.fastq.g
```

```
## Encountered 904 unique sequences from 2914 total sequences read.

## Dereplicating sequence entries in Fastq file: /home/rstudio/MiSeq_SOP/filtered/F3D143_F_filt.fastq.g

## Encountered 939 unique sequences from 2941 total sequences read.

## Dereplicating sequence entries in Fastq file: /home/rstudio/MiSeq_SOP/filtered/F3D144_F_filt.fastq.g

## Encountered 1267 unique sequences from 4312 total sequences read.

## Dereplicating sequence entries in Fastq file: /home/rstudio/MiSeq_SOP/filtered/F3D145_F_filt.fastq.g

## Encountered 1756 unique sequences from 6741 total sequences read.

## Dereplicating sequence entries in Fastq file: /home/rstudio/MiSeq_SOP/filtered/F3D146_F_filt.fastq.g

## Encountered 1438 unique sequences from 4560 total sequences read.

## Dereplicating sequence entries in Fastq file: /home/rstudio/MiSeq_SOP/filtered/F3D147_F_filt.fastq.g

## Encountered 3590 unique sequences from 15637 total sequences read.

## Dereplicating sequence entries in Fastq file: /home/rstudio/MiSeq_SOP/filtered/F3D148_F_filt.fastq.g

## Encountered 2762 unique sequences from 11413 total sequences read.

## Dereplicating sequence entries in Fastq file: /home/rstudio/MiSeq_SOP/filtered/F3D149_F_filt.fastq.g

## Encountered 3021 unique sequences from 12017 total sequences read.

## Dereplicating sequence entries in Fastq file: /home/rstudio/MiSeq_SOP/filtered/F3D150_F_filt.fastq.g

## Encountered 1566 unique sequences from 5032 total sequences read.

## Dereplicating sequence entries in Fastq file: /home/rstudio/MiSeq_SOP/filtered/F3D2_F_filt.fastq.gz

## Encountered 3707 unique sequences from 18075 total sequences read.

## Dereplicating sequence entries in Fastq file: /home/rstudio/MiSeq_SOP/filtered/F3D3_F_filt.fastq.gz

## Encountered 1479 unique sequences from 6250 total sequences read.

## Dereplicating sequence entries in Fastq file: /home/rstudio/MiSeq_SOP/filtered/F3D5_F_filt.fastq.gz

## Encountered 1195 unique sequences from 4052 total sequences read.

## Dereplicating sequence entries in Fastq file: /home/rstudio/MiSeq_SOP/filtered/F3D6_F_filt.fastq.gz
```

```

## Encountered 1832 unique sequences from 7369 total sequences read.

## Dereplicating sequence entries in Fastq file: /home/rstudio/MiSeq_SOP/filtered/F3D7_F_filt.fastq.gz

## Encountered 1183 unique sequences from 4765 total sequences read.

## Dereplicating sequence entries in Fastq file: /home/rstudio/MiSeq_SOP/filtered/F3D8_F_filt.fastq.gz

## Encountered 1382 unique sequences from 4871 total sequences read.

## Dereplicating sequence entries in Fastq file: /home/rstudio/MiSeq_SOP/filtered/F3D9_F_filt.fastq.gz

## Encountered 1709 unique sequences from 6504 total sequences read.

## Dereplicating sequence entries in Fastq file: /home/rstudio/MiSeq_SOP/filtered/Mock_F_filt.fastq.gz

## Encountered 897 unique sequences from 4314 total sequences read.

derepRs <- derepFastq(filtRs, verbose=TRUE)

## Dereplicating sequence entries in Fastq file: /home/rstudio/MiSeq_SOP/filtered/F3D0_R_filt.fastq.gz

## Encountered 1660 unique sequences from 7113 total sequences read.

## Dereplicating sequence entries in Fastq file: /home/rstudio/MiSeq_SOP/filtered/F3D1_R_filt.fastq.gz

## Encountered 1349 unique sequences from 5299 total sequences read.

## Dereplicating sequence entries in Fastq file: /home/rstudio/MiSeq_SOP/filtered/F3D141_R_filt.fastq.gz

## Encountered 1335 unique sequences from 5463 total sequences read.

## Dereplicating sequence entries in Fastq file: /home/rstudio/MiSeq_SOP/filtered/F3D142_R_filt.fastq.gz

## Encountered 853 unique sequences from 2914 total sequences read.

## Dereplicating sequence entries in Fastq file: /home/rstudio/MiSeq_SOP/filtered/F3D143_R_filt.fastq.gz

## Encountered 880 unique sequences from 2941 total sequences read.

## Dereplicating sequence entries in Fastq file: /home/rstudio/MiSeq_SOP/filtered/F3D144_R_filt.fastq.gz

## Encountered 1286 unique sequences from 4312 total sequences read.

## Dereplicating sequence entries in Fastq file: /home/rstudio/MiSeq_SOP/filtered/F3D145_R_filt.fastq.gz

## Encountered 1803 unique sequences from 6741 total sequences read.

```



```
## Dereplicating sequence entries in Fastq file: /home/rstudio/MiSeq_SOP/filtered/F3D146_R_filt.fastq.gz
## Encountered 1265 unique sequences from 4560 total sequences read.
## Dereplicating sequence entries in Fastq file: /home/rstudio/MiSeq_SOP/filtered/F3D147_R_filt.fastq.gz
## Encountered 3414 unique sequences from 15637 total sequences read.
## Dereplicating sequence entries in Fastq file: /home/rstudio/MiSeq_SOP/filtered/F3D148_R_filt.fastq.gz
## Encountered 2522 unique sequences from 11413 total sequences read.
## Dereplicating sequence entries in Fastq file: /home/rstudio/MiSeq_SOP/filtered/F3D149_R_filt.fastq.gz
## Encountered 2771 unique sequences from 12017 total sequences read.
## Dereplicating sequence entries in Fastq file: /home/rstudio/MiSeq_SOP/filtered/F3D150_R_filt.fastq.gz
## Encountered 1415 unique sequences from 5032 total sequences read.
## Dereplicating sequence entries in Fastq file: /home/rstudio/MiSeq_SOP/filtered/F3D2_R_filt.fastq.gz
## Encountered 3290 unique sequences from 18075 total sequences read.
## Dereplicating sequence entries in Fastq file: /home/rstudio/MiSeq_SOP/filtered/F3D3_R_filt.fastq.gz
## Encountered 1390 unique sequences from 6250 total sequences read.
## Dereplicating sequence entries in Fastq file: /home/rstudio/MiSeq_SOP/filtered/F3D5_R_filt.fastq.gz
## Encountered 1134 unique sequences from 4052 total sequences read.
## Dereplicating sequence entries in Fastq file: /home/rstudio/MiSeq_SOP/filtered/F3D6_R_filt.fastq.gz
## Encountered 1635 unique sequences from 7369 total sequences read.
## Dereplicating sequence entries in Fastq file: /home/rstudio/MiSeq_SOP/filtered/F3D7_R_filt.fastq.gz
## Encountered 1084 unique sequences from 4765 total sequences read.
## Dereplicating sequence entries in Fastq file: /home/rstudio/MiSeq_SOP/filtered/F3D8_R_filt.fastq.gz
## Encountered 1161 unique sequences from 4871 total sequences read.
## Dereplicating sequence entries in Fastq file: /home/rstudio/MiSeq_SOP/filtered/F3D9_R_filt.fastq.gz
## Encountered 1502 unique sequences from 6504 total sequences read.
## Dereplicating sequence entries in Fastq file: /home/rstudio/MiSeq_SOP/filtered/Mock_R_filt.fastq.gz
## Encountered 732 unique sequences from 4314 total sequences read.
```

```
names(derepFs) <- sampleNames
names(derepRs) <- sampleNames
```

Les taux d'erreurs

```
errF <- learnErrors(filtFs, multithread=TRUE)
```

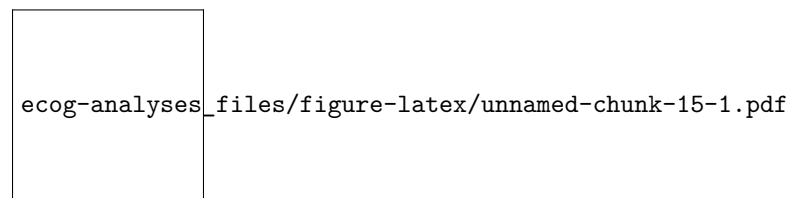
33514080 total bases in 139642 reads from 20 samples will be used for learning the error rates.

```
errR <- learnErrors(filtRs, multithread=TRUE)
```

22342720 total bases in 139642 reads from 20 samples will be used for learning the error rates.

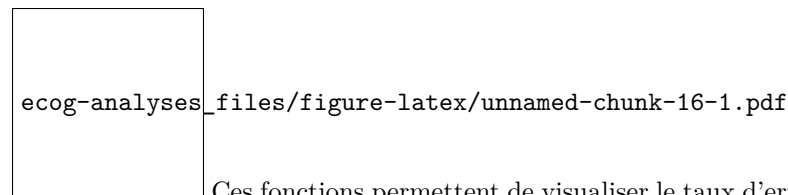
```
plotErrors(errF)
```

Warning: Transformation introduced infinite values in continuous y-axis



```
plotErrors(errR)
```

Warning: Transformation introduced infinite values in continuous y-axis



Ces fonctions permettent de visualiser le taux d'erreur estimés avec learnErrors (correspondant à la ligne noire) et le taux d'erreur observé représenté par la ligne rouge.

Inférence d'échantillon

```
dadaFs <- dada(derepFs, err=errF, multithread=TRUE)
```

```
## Sample 1 - 7113 reads in 1979 unique sequences.
## Sample 2 - 5299 reads in 1639 unique sequences.
## Sample 3 - 5463 reads in 1477 unique sequences.
## Sample 4 - 2914 reads in 904 unique sequences.
## Sample 5 - 2941 reads in 939 unique sequences.
## Sample 6 - 4312 reads in 1267 unique sequences.
## Sample 7 - 6741 reads in 1756 unique sequences.
## Sample 8 - 4560 reads in 1438 unique sequences.
## Sample 9 - 15637 reads in 3590 unique sequences.
```

```
## Sample 10 - 11413 reads in 2762 unique sequences.
## Sample 11 - 12017 reads in 3021 unique sequences.
## Sample 12 - 5032 reads in 1566 unique sequences.
## Sample 13 - 18075 reads in 3707 unique sequences.
## Sample 14 - 6250 reads in 1479 unique sequences.
## Sample 15 - 4052 reads in 1195 unique sequences.
## Sample 16 - 7369 reads in 1832 unique sequences.
## Sample 17 - 4765 reads in 1183 unique sequences.
## Sample 18 - 4871 reads in 1382 unique sequences.
## Sample 19 - 6504 reads in 1709 unique sequences.
## Sample 20 - 4314 reads in 897 unique sequences.
```

```
dadaRs <- dada(derepRs, err=errR, multithread=TRUE)
```

```
## Sample 1 - 7113 reads in 1660 unique sequences.
## Sample 2 - 5299 reads in 1349 unique sequences.
## Sample 3 - 5463 reads in 1335 unique sequences.
## Sample 4 - 2914 reads in 853 unique sequences.
## Sample 5 - 2941 reads in 880 unique sequences.
## Sample 6 - 4312 reads in 1286 unique sequences.
## Sample 7 - 6741 reads in 1803 unique sequences.
## Sample 8 - 4560 reads in 1265 unique sequences.
## Sample 9 - 15637 reads in 3414 unique sequences.
## Sample 10 - 11413 reads in 2522 unique sequences.
## Sample 11 - 12017 reads in 2771 unique sequences.
## Sample 12 - 5032 reads in 1415 unique sequences.
## Sample 13 - 18075 reads in 3290 unique sequences.
## Sample 14 - 6250 reads in 1390 unique sequences.
## Sample 15 - 4052 reads in 1134 unique sequences.
## Sample 16 - 7369 reads in 1635 unique sequences.
## Sample 17 - 4765 reads in 1084 unique sequences.
## Sample 18 - 4871 reads in 1161 unique sequences.
## Sample 19 - 6504 reads in 1502 unique sequences.
## Sample 20 - 4314 reads in 732 unique sequences.
```

```
dadaFs[[1]]
```

```
## dada-class: object describing DADA2 denoising results
## 128 sequence variants were inferred from 1979 input unique sequences.
## Key parameters: OMEGA_A = 1e-40, OMEGA_C = 1e-40, BAND_SIZE = 16
```

```
#Construire une table de séquence et supprimer des chimères
```

```
##Alignement des R1 et R2 en contigs
```

```
mergers <- mergePairs(dadaFs, derepFs, dadaRs, derepRs)
```

```
##Construction de la table d'observation
```

```
seqtabAll <- makeSequenceTable(mergers[!grepl("Mock", names(mergers))])
table(nchar(getSequences(seqtabAll)))
```

```
##
## 251 252 253 254 255
##   1  85 186   5   2
```

##Détection des chimères

```
seqtabNoC <- removeBimeraDenovo(seqtabAll)
```

Cette détection permet d'avoir une meilleure analyse, en éliminant les chevauchements de séquences qui fausse les résultats. Pour visualiser sous forme de pourcentage le taux de chimère on peut utiliser cette formule :

```
1-sum(seqtabNoC)/sum(seqtabAll)
```

```
## [1] 0.03719456
```

#**Assignation taxonomique** L'utilisation de wget permet de télécharger les données directement à partir d'un lien internet.

```
cd ~
wget https://zenodo.org/record/4587955/files/silva_nr99_v138.1_train_set.fa.gz
```

```
## --2021-12-25 15:24:59-- https://zenodo.org/record/4587955/files/silva_nr99_v138.1_train_set.fa.gz
## Resolving zenodo.org (zenodo.org)... 137.138.76.77
## Connecting to zenodo.org (zenodo.org)|137.138.76.77|:443... connected.
## HTTP request sent, awaiting response... 200 OK
## Length: 137283333 (131M) [application/octet-stream]
## Saving to: 'silva_nr99_v138.1_train_set.fa.gz.6'
##
##      OK ..... 0% 14.8M 9s
##      50K ..... 0% 7.01M 14s
##      100K ..... 0% 13.1M 13s
##      150K ..... 0% 10.4M 13s
##      200K ..... 0% 93.6M 10s
##      250K ..... 0% 10.3M 11s
##      300K ..... 0% 79.6M 9s
##      350K ..... 0% 84.9M 8s
##      400K ..... 0% 98.6M 8s
##      450K ..... 0% 13.8M 8s
##      500K ..... 0% 94.1M 7s
##      550K ..... 0% 83.5M 7s
##      600K ..... 0% 79.2M 6s
##      650K ..... 0% 78.3M 6s
##      700K ..... 0% 97.8M 6s
##      750K ..... 0% 88.7M 5s
##      800K ..... 0% 101M 5s
##      850K ..... 0% 91.9M 5s
##      900K ..... 0% 42.4M 5s
##      950K ..... 0% 64.1M 5s
##     1000K ..... 0% 48.1M 5s
##     1050K ..... 0% 101M 4s
##     1100K ..... 0% 109M 4s
```

##	1150K	0%	65.5M	4s
##	1200K	0%	87.9M	4s
##	1250K	0%	106M	4s
##	1300K	1%	95.3M	4s
##	1350K	1%	15.8M	4s
##	1400K	1%	104M	4s
##	1450K	1%	103M	4s
##	1500K	1%	96.7M	4s
##	1550K	1%	28.5M	4s
##	1600K	1%	103M	4s
##	1650K	1%	114M	4s
##	1700K	1%	109M	4s
##	1750K	1%	93.6M	4s
##	1800K	1%	87.6M	3s
##	1850K	1%	120M	3s
##	1900K	1%	81.2M	3s
##	1950K	1%	73.3M	3s
##	2000K	1%	97.9M	3s
##	2050K	1%	86.6M	3s
##	2100K	1%	92.1M	3s
##	2150K	1%	85.0M	3s
##	2200K	1%	92.6M	3s
##	2250K	1%	88.6M	3s
##	2300K	1%	73.0M	3s
##	2350K	1%	72.6M	3s
##	2400K	1%	113M	3s
##	2450K	1%	87.4M	3s
##	2500K	1%	83.6M	3s
##	2550K	1%	84.8M	3s
##	2600K	1%	102M	3s
##	2650K	2%	108M	3s
##	2700K	2%	79.5M	3s
##	2750K	2%	65.9M	3s
##	2800K	2%	93.8M	3s
##	2850K	2%	91.2M	3s
##	2900K	2%	90.5M	3s
##	2950K	2%	59.9M	3s
##	3000K	2%	92.3M	3s
##	3050K	2%	77.9M	3s
##	3100K	2%	40.2M	3s
##	3150K	2%	68.4M	3s
##	3200K	2%	81.8M	3s
##	3250K	2%	78.7M	3s
##	3300K	2%	75.7M	3s
##	3350K	2%	59.4M	3s
##	3400K	2%	75.5M	3s
##	3450K	2%	78.0M	3s
##	3500K	2%	79.7M	3s
##	3550K	2%	70.3M	3s
##	3600K	2%	78.7M	3s
##	3650K	2%	90.0M	3s
##	3700K	2%	43.6M	3s
##	3750K	2%	54.8M	3s
##	3800K	2%	77.0M	2s

##	3850K	2%	77.1M	2s
##	3900K	2%	74.1M	2s
##	3950K	2%	64.4M	2s
##	4000K	3%	81.0M	2s
##	4050K	3%	54.2M	2s
##	4100K	3%	73.4M	2s
##	4150K	3%	28.5M	2s
##	4200K	3%	101M	2s
##	4250K	3%	12.8M	3s
##	4300K	3%	90.9M	3s
##	4350K	3%	91.3M	3s
##	4400K	3%	103M	2s
##	4450K	3%	106M	2s
##	4500K	3%	98.3M	2s
##	4550K	3%	70.0M	2s
##	4600K	3%	97.3M	2s
##	4650K	3%	104M	2s
##	4700K	3%	110M	2s
##	4750K	3%	84.9M	2s
##	4800K	3%	103M	2s
##	4850K	3%	58.7M	2s
##	4900K	3%	104M	2s
##	4950K	3%	76.7M	2s
##	5000K	3%	113M	2s
##	5050K	3%	97.0M	2s
##	5100K	3%	110M	2s
##	5150K	3%	91.5M	2s
##	5200K	3%	100M	2s
##	5250K	3%	101M	2s
##	5300K	3%	109M	2s
##	5350K	4%	85.1M	2s
##	5400K	4%	93.3M	2s
##	5450K	4%	83.6M	2s
##	5500K	4%	82.2M	2s
##	5550K	4%	55.0M	2s
##	5600K	4%	84.4M	2s
##	5650K	4%	99.4M	2s
##	5700K	4%	99.2M	2s
##	5750K	4%	106M	2s
##	5800K	4%	94.5M	2s
##	5850K	4%	94.0M	2s
##	5900K	4%	104M	2s
##	5950K	4%	72.0M	2s
##	6000K	4%	82.8M	2s
##	6050K	4%	91.7M	2s
##	6100K	4%	103M	2s
##	6150K	4%	80.3M	2s
##	6200K	4%	76.0M	2s
##	6250K	4%	81.0M	2s
##	6300K	4%	78.8M	2s
##	6350K	4%	73.5M	2s
##	6400K	4%	78.0M	2s
##	6450K	4%	82.5M	2s
##	6500K	4%	83.9M	2s

##	6550K	4%	72.8M	2s
##	6600K	4%	89.6M	2s
##	6650K	4%	78.9M	2s
##	6700K	5%	90.3M	2s
##	6750K	5%	77.5M	2s
##	6800K	5%	98.1M	2s
##	6850K	5%	58.6M	2s
##	6900K	5%	83.2M	2s
##	6950K	5%	77.9M	2s
##	7000K	5%	94.0M	2s
##	7050K	5%	88.1M	2s
##	7100K	5%	78.5M	2s
##	7150K	5%	75.4M	2s
##	7200K	5%	101M	2s
##	7250K	5%	56.7M	2s
##	7300K	5%	71.6M	2s
##	7350K	5%	79.0M	2s
##	7400K	5%	103M	2s
##	7450K	5%	104M	2s
##	7500K	5%	59.9M	2s
##	7550K	5%	75.6M	2s
##	7600K	5%	108M	2s
##	7650K	5%	89.1M	2s
##	7700K	5%	104M	2s
##	7750K	5%	90.5M	2s
##	7800K	5%	43.9M	2s
##	7850K	5%	87.1M	2s
##	7900K	5%	107M	2s
##	7950K	5%	79.8M	2s
##	8000K	6%	67.1M	2s
##	8050K	6%	117M	2s
##	8100K	6%	90.1M	2s
##	8150K	6%	62.3M	2s
##	8200K	6%	112M	2s
##	8250K	6%	96.2M	2s
##	8300K	6%	89.8M	2s
##	8350K	6%	89.8M	2s
##	8400K	6%	111M	2s
##	8450K	6%	73.9M	2s
##	8500K	6%	87.9M	2s
##	8550K	6%	92.0M	2s
##	8600K	6%	100M	2s
##	8650K	6%	84.2M	2s
##	8700K	6%	105M	2s
##	8750K	6%	62.0M	2s
##	8800K	6%	90.3M	2s
##	8850K	6%	102M	2s
##	8900K	6%	113M	2s
##	8950K	6%	93.6M	2s
##	9000K	6%	102M	2s
##	9050K	6%	112M	2s
##	9100K	6%	60.5M	2s
##	9150K	6%	76.5M	2s
##	9200K	6%	96.2M	2s

##	9250K	6%	95.6M	2s
##	9300K	6%	99.6M	2s
##	9350K	7%	99.4M	2s
##	9400K	7%	65.4M	2s
##	9450K	7%	89.5M	2s
##	9500K	7%	102M	2s
##	9550K	7%	87.6M	2s
##	9600K	7%	94.3M	2s
##	9650K	7%	127M	2s
##	9700K	7%	86.5M	2s
##	9750K	7%	77.8M	2s
##	9800K	7%	114M	2s
##	9850K	7%	98.6M	2s
##	9900K	7%	93.7M	2s
##	9950K	7%	91.6M	2s
##	10000K	7%	104M	2s
##	10050K	7%	100M	2s
##	10100K	7%	102M	2s
##	10150K	7%	81.4M	2s
##	10200K	7%	100M	2s
##	10250K	7%	97.2M	2s
##	10300K	7%	128M	2s
##	10350K	7%	52.5M	2s
##	10400K	7%	103M	2s
##	10450K	7%	137M	2s
##	10500K	7%	93.6M	2s
##	10550K	7%	85.3M	2s
##	10600K	7%	108M	2s
##	10650K	7%	119M	2s
##	10700K	8%	107M	2s
##	10750K	8%	68.2M	2s
##	10800K	8%	110M	2s
##	10850K	8%	93.6M	2s
##	10900K	8%	121M	2s
##	10950K	8%	116M	2s
##	11000K	8%	81.4M	2s
##	11050K	8%	106M	2s
##	11100K	8%	102M	2s
##	11150K	8%	85.0M	2s
##	11200K	8%	102M	2s
##	11250K	8%	134M	2s
##	11300K	8%	75.1M	2s
##	11350K	8%	88.3M	2s
##	11400K	8%	114M	2s
##	11450K	8%	94.8M	2s
##	11500K	8%	102M	2s
##	11550K	8%	105M	2s
##	11600K	8%	113M	2s
##	11650K	8%	98.4M	2s
##	11700K	8%	112M	2s
##	11750K	8%	83.9M	2s
##	11800K	8%	101M	2s
##	11850K	8%	113M	2s
##	11900K	8%	134M	2s

##	11950K	8%	83.4M	2s
##	12000K	8%	95.5M	2s
##	12050K	9%	109M	2s
##	12100K	9%	95.9M	2s
##	12150K	9%	88.2M	2s
##	12200K	9%	132M	2s
##	12250K	9%	89.3M	2s
##	12300K	9%	97.5M	2s
##	12350K	9%	89.2M	2s
##	12400K	9%	98.1M	2s
##	12450K	9%	109M	2s
##	12500K	9%	130M	2s
##	12550K	9%	106M	2s
##	12600K	9%	101M	2s
##	12650K	9%	108M	2s
##	12700K	9%	98.7M	2s
##	12750K	9%	79.2M	2s
##	12800K	9%	132M	2s
##	12850K	9%	131M	2s
##	12900K	9%	75.0M	2s
##	12950K	9%	90.7M	2s
##	13000K	9%	91.6M	2s
##	13050K	9%	106M	2s
##	13100K	9%	136M	2s
##	13150K	9%	89.1M	2s
##	13200K	9%	116M	2s
##	13250K	9%	102M	2s
##	13300K	9%	107M	2s
##	13350K	9%	84.2M	2s
##	13400K	10%	118M	2s
##	13450K	10%	114M	2s
##	13500K	10%	109M	2s
##	13550K	10%	79.7M	2s
##	13600K	10%	92.9M	2s
##	13650K	10%	102M	2s
##	13700K	10%	120M	2s
##	13750K	10%	95.4M	2s
##	13800K	10%	117M	2s
##	13850K	10%	7.19M	2s
##	13900K	10%	139M	2s
##	13950K	10%	108M	2s
##	14000K	10%	140M	2s
##	14050K	10%	131M	2s
##	14100K	10%	130M	2s
##	14150K	10%	121M	2s
##	14200K	10%	130M	2s
##	14250K	10%	139M	2s
##	14300K	10%	134M	2s
##	14350K	10%	112M	2s
##	14400K	10%	111M	2s
##	14450K	10%	123M	2s
##	14500K	10%	137M	2s
##	14550K	10%	94.9M	2s
##	14600K	10%	125M	2s

##	14650K	10%	123M	2s
##	14700K	11%	117M	2s
##	14750K	11%	114M	2s
##	14800K	11%	139M	2s
##	14850K	11%	52.2M	2s
##	14900K	11%	121M	2s
##	14950K	11%	114M	2s
##	15000K	11%	133M	2s
##	15050K	11%	137M	2s
##	15100K	11%	41.3M	2s
##	15150K	11%	88.3M	2s
##	15200K	11%	110M	2s
##	15250K	11%	98.4M	2s
##	15300K	11%	105M	2s
##	15350K	11%	79.8M	2s
##	15400K	11%	100M	2s
##	15450K	11%	126M	2s
##	15500K	11%	111M	2s
##	15550K	11%	98.6M	2s
##	15600K	11%	128M	2s
##	15650K	11%	124M	2s
##	15700K	11%	125M	2s
##	15750K	11%	95.6M	2s
##	15800K	11%	110M	2s
##	15850K	11%	119M	2s
##	15900K	11%	115M	2s
##	15950K	11%	85.6M	2s
##	16000K	11%	96.8M	2s
##	16050K	12%	102M	2s
##	16100K	12%	105M	2s
##	16150K	12%	90.4M	2s
##	16200K	12%	104M	2s
##	16250K	12%	118M	2s
##	16300K	12%	101M	2s
##	16350K	12%	101M	2s
##	16400K	12%	122M	2s
##	16450K	12%	101M	2s
##	16500K	12%	117M	2s
##	16550K	12%	112M	2s
##	16600K	12%	98.8M	2s
##	16650K	12%	118M	2s
##	16700K	12%	116M	2s
##	16750K	12%	113M	2s
##	16800K	12%	98.4M	2s
##	16850K	12%	114M	2s
##	16900K	12%	119M	2s
##	16950K	12%	88.6M	2s
##	17000K	12%	108M	2s
##	17050K	12%	120M	2s
##	17100K	12%	99.6M	2s
##	17150K	12%	91.3M	2s
##	17200K	12%	106M	2s
##	17250K	12%	111M	2s
##	17300K	12%	108M	2s

##	17350K	12%	88.8M	2s
##	17400K	13%	111M	2s
##	17450K	13%	117M	2s
##	17500K	13%	119M	2s
##	17550K	13%	95.5M	1s
##	17600K	13%	72.2M	1s
##	17650K	13%	81.9M	1s
##	17700K	13%	94.8M	1s
##	17750K	13%	43.1M	1s
##	17800K	13%	89.6M	1s
##	17850K	13%	101M	1s
##	17900K	13%	111M	1s
##	17950K	13%	93.8M	1s
##	18000K	13%	109M	1s
##	18050K	13%	109M	1s
##	18100K	13%	100M	1s
##	18150K	13%	90.2M	1s
##	18200K	13%	116M	1s
##	18250K	13%	104M	1s
##	18300K	13%	128M	1s
##	18350K	13%	87.0M	1s
##	18400K	13%	105M	1s
##	18450K	13%	100M	1s
##	18500K	13%	94.9M	1s
##	18550K	13%	96.7M	1s
##	18600K	13%	100M	1s
##	18650K	13%	111M	1s
##	18700K	13%	105M	1s
##	18750K	14%	88.0M	1s
##	18800K	14%	116M	1s
##	18850K	14%	110M	1s
##	18900K	14%	107M	1s
##	18950K	14%	92.2M	1s
##	19000K	14%	102M	1s
##	19050K	14%	117M	1s
##	19100K	14%	95.4M	1s
##	19150K	14%	96.5M	1s
##	19200K	14%	110M	1s
##	19250K	14%	105M	1s
##	19300K	14%	95.0M	1s
##	19350K	14%	107M	1s
##	19400K	14%	112M	1s
##	19450K	14%	102M	1s
##	19500K	14%	118M	1s
##	19550K	14%	85.8M	1s
##	19600K	14%	105M	1s
##	19650K	14%	115M	1s
##	19700K	14%	110M	1s
##	19750K	14%	84.2M	1s
##	19800K	14%	104M	1s
##	19850K	14%	87.6M	1s
##	19900K	14%	90.4M	1s
##	19950K	14%	87.7M	1s
##	20000K	14%	125M	1s

##	20050K	14%	113M	1s
##	20100K	15%	116M	1s
##	20150K	15%	101M	1s
##	20200K	15%	94.4M	1s
##	20250K	15%	105M	1s
##	20300K	15%	133M	1s
##	20350K	15%	77.4M	1s
##	20400K	15%	109M	1s
##	20450K	15%	108M	1s
##	20500K	15%	101M	1s
##	20550K	15%	96.0M	1s
##	20600K	15%	114M	1s
##	20650K	15%	107M	1s
##	20700K	15%	101M	1s
##	20750K	15%	81.4M	1s
##	20800K	15%	96.7M	1s
##	20850K	15%	91.1M	1s
##	20900K	15%	109M	1s
##	20950K	15%	119M	1s
##	21000K	15%	95.3M	1s
##	21050K	15%	109M	1s
##	21100K	15%	109M	1s
##	21150K	15%	77.3M	1s
##	21200K	15%	116M	1s
##	21250K	15%	124M	1s
##	21300K	15%	103M	1s
##	21350K	15%	97.1M	1s
##	21400K	15%	96.6M	1s
##	21450K	16%	109M	1s
##	21500K	16%	122M	1s
##	21550K	16%	72.1M	1s
##	21600K	16%	94.1M	1s
##	21650K	16%	96.9M	1s
##	21700K	16%	105M	1s
##	21750K	16%	91.0M	1s
##	21800K	16%	113M	1s
##	21850K	16%	137M	1s
##	21900K	16%	106M	1s
##	21950K	16%	48.5M	1s
##	22000K	16%	92.0M	1s
##	22050K	16%	96.4M	1s
##	22100K	16%	98.0M	1s
##	22150K	16%	96.2M	1s
##	22200K	16%	121M	1s
##	22250K	16%	105M	1s
##	22300K	16%	96.4M	1s
##	22350K	16%	100M	1s
##	22400K	16%	107M	1s
##	22450K	16%	121M	1s
##	22500K	16%	108M	1s
##	22550K	16%	87.2M	1s
##	22600K	16%	98.2M	1s
##	22650K	16%	111M	1s
##	22700K	16%	71.7M	1s

##	22750K	17%	96.7M	1s
##	22800K	17%	121M	1s
##	22850K	17%	98.6M	1s
##	22900K	17%	104M	1s
##	22950K	17%	110M	1s
##	23000K	17%	113M	1s
##	23050K	17%	102M	1s
##	23100K	17%	100M	1s
##	23150K	17%	89.1M	1s
##	23200K	17%	107M	1s
##	23250K	17%	114M	1s
##	23300K	17%	115M	1s
##	23350K	17%	93.7M	1s
##	23400K	17%	106M	1s
##	23450K	17%	95.0M	1s
##	23500K	17%	97.9M	1s
##	23550K	17%	104M	1s
##	23600K	17%	108M	1s
##	23650K	17%	98.3M	1s
##	23700K	17%	103M	1s
##	23750K	17%	86.5M	1s
##	23800K	17%	122M	1s
##	23850K	17%	98.3M	1s
##	23900K	17%	110M	1s
##	23950K	17%	95.3M	1s
##	24000K	17%	94.8M	1s
##	24050K	17%	95.7M	1s
##	24100K	18%	98.5M	1s
##	24150K	18%	113M	1s
##	24200K	18%	116M	1s
##	24250K	18%	104M	1s
##	24300K	18%	111M	1s
##	24350K	18%	98.1M	1s
##	24400K	18%	101M	1s
##	24450K	18%	96.5M	1s
##	24500K	18%	121M	1s
##	24550K	18%	90.3M	1s
##	24600K	18%	109M	1s
##	24650K	18%	99.0M	1s
##	24700K	18%	87.2M	1s
##	24750K	18%	94.9M	1s
##	24800K	18%	73.1M	1s
##	24850K	18%	94.3M	1s
##	24900K	18%	94.7M	1s
##	24950K	18%	102M	1s
##	25000K	18%	98.6M	1s
##	25050K	18%	109M	1s
##	25100K	18%	140M	1s
##	25150K	18%	87.2M	1s
##	25200K	18%	103M	1s
##	25250K	18%	96.9M	1s
##	25300K	18%	135M	1s
##	25350K	18%	107M	1s
##	25400K	18%	103M	1s

##	25450K	19%	111M	1s
##	25500K	19%	107M	1s
##	25550K	19%	101M	1s
##	25600K	19%	102M	1s
##	25650K	19%	118M	1s
##	25700K	19%	129M	1s
##	25750K	19%	92.8M	1s
##	25800K	19%	117M	1s
##	25850K	19%	84.5M	1s
##	25900K	19%	138M	1s
##	25950K	19%	113M	1s
##	26000K	19%	111M	1s
##	26050K	19%	134M	1s
##	26100K	19%	123M	1s
##	26150K	19%	92.0M	1s
##	26200K	19%	137M	1s
##	26250K	19%	137M	1s
##	26300K	19%	105M	1s
##	26350K	19%	105M	1s
##	26400K	19%	77.4M	1s
##	26450K	19%	132M	1s
##	26500K	19%	134M	1s
##	26550K	19%	121M	1s
##	26600K	19%	50.9M	1s
##	26650K	19%	135M	1s
##	26700K	19%	132M	1s
##	26750K	19%	98.9M	1s
##	26800K	20%	119M	1s
##	26850K	20%	133M	1s
##	26900K	20%	132M	1s
##	26950K	20%	118M	1s
##	27000K	20%	135M	1s
##	27050K	20%	128M	1s
##	27100K	20%	128M	1s
##	27150K	20%	119M	1s
##	27200K	20%	135M	1s
##	27250K	20%	130M	1s
##	27300K	20%	136M	1s
##	27350K	20%	110M	1s
##	27400K	20%	125M	1s
##	27450K	20%	114M	1s
##	27500K	20%	132M	1s
##	27550K	20%	109M	1s
##	27600K	20%	131M	1s
##	27650K	20%	139M	1s
##	27700K	20%	133M	1s
##	27750K	20%	117M	1s
##	27800K	20%	137M	1s
##	27850K	20%	125M	1s
##	27900K	20%	119M	1s
##	27950K	20%	105M	1s
##	28000K	20%	131M	1s
##	28050K	20%	131M	1s
##	28100K	20%	132M	1s

##	28150K	21%	105M	1s
##	28200K	21%	123M	1s
##	28250K	21%	75.5M	1s
##	28300K	21%	138M	1s
##	28350K	21%	88.0M	1s
##	28400K	21%	130M	1s
##	28450K	21%	110M	1s
##	28500K	21%	112M	1s
##	28550K	21%	88.0M	1s
##	28600K	21%	119M	1s
##	28650K	21%	117M	1s
##	28700K	21%	100M	1s
##	28750K	21%	94.7M	1s
##	28800K	21%	100M	1s
##	28850K	21%	99.5M	1s
##	28900K	21%	97.2M	1s
##	28950K	21%	101M	1s
##	29000K	21%	120M	1s
##	29050K	21%	140M	1s
##	29100K	21%	49.7M	1s
##	29150K	21%	54.3M	1s
##	29200K	21%	125M	1s
##	29250K	21%	46.0M	1s
##	29300K	21%	104M	1s
##	29350K	21%	51.2M	1s
##	29400K	21%	97.7M	1s
##	29450K	22%	46.4M	1s
##	29500K	22%	101M	1s
##	29550K	22%	76.5M	1s
##	29600K	22%	83.5M	1s
##	29650K	22%	98.4M	1s
##	29700K	22%	51.0M	1s
##	29750K	22%	91.3M	1s
##	29800K	22%	75.9M	1s
##	29850K	22%	92.1M	1s
##	29900K	22%	49.2M	1s
##	29950K	22%	77.3M	1s
##	30000K	22%	98.4M	1s
##	30050K	22%	86.8M	1s
##	30100K	22%	54.9M	1s
##	30150K	22%	73.9M	1s
##	30200K	22%	85.0M	1s
##	30250K	22%	97.1M	1s
##	30300K	22%	101M	1s
##	30350K	22%	44.2M	1s
##	30400K	22%	79.1M	1s
##	30450K	22%	92.6M	1s
##	30500K	22%	90.1M	1s
##	30550K	22%	70.9M	1s
##	30600K	22%	72.5M	1s
##	30650K	22%	101M	1s
##	30700K	22%	104M	1s
##	30750K	22%	53.7M	1s
##	30800K	23%	81.6M	1s

##	30850K	23%	98.1M	1s
##	30900K	23%	103M	1s
##	30950K	23%	80.9M	1s
##	31000K	23%	64.2M	1s
##	31050K	23%	80.8M	1s
##	31100K	23%	95.5M	1s
##	31150K	23%	58.0M	1s
##	31200K	23%	93.9M	1s
##	31250K	23%	112M	1s
##	31300K	23%	74.6M	1s
##	31350K	23%	43.7M	1s
##	31400K	23%	81.1M	1s
##	31450K	23%	101M	1s
##	31500K	23%	117M	1s
##	31550K	23%	88.1M	1s
##	31600K	23%	86.5M	1s
##	31650K	23%	108M	1s
##	31700K	23%	73.5M	1s
##	31750K	23%	93.6M	1s
##	31800K	23%	83.4M	1s
##	31850K	23%	94.7M	1s
##	31900K	23%	89.4M	1s
##	31950K	23%	98.9M	1s
##	32000K	23%	64.4M	1s
##	32050K	23%	64.1M	1s
##	32100K	23%	95.8M	1s
##	32150K	24%	98.6M	1s
##	32200K	24%	105M	1s
##	32250K	24%	111M	1s
##	32300K	24%	114M	1s
##	32350K	24%	56.3M	1s
##	32400K	24%	100M	1s
##	32450K	24%	66.8M	1s
##	32500K	24%	99.5M	1s
##	32550K	24%	120M	1s
##	32600K	24%	103M	1s
##	32650K	24%	55.1M	1s
##	32700K	24%	96.2M	1s
##	32750K	24%	89.3M	1s
##	32800K	24%	103M	1s
##	32850K	24%	121M	1s
##	32900K	24%	109M	1s
##	32950K	24%	111M	1s
##	33000K	24%	72.4M	1s
##	33050K	24%	79.7M	1s
##	33100K	24%	92.9M	1s
##	33150K	24%	74.8M	1s
##	33200K	24%	97.9M	1s
##	33250K	24%	102M	1s
##	33300K	24%	113M	1s
##	33350K	24%	65.2M	1s
##	33400K	24%	91.3M	1s
##	33450K	24%	101M	1s
##	33500K	25%	106M	1s

##	33550K	25%	77.3M	1s
##	33600K	25%	110M	1s
##	33650K	25%	88.5M	1s
##	33700K	25%	91.1M	1s
##	33750K	25%	96.5M	1s
##	33800K	25%	81.5M	1s
##	33850K	25%	117M	1s
##	33900K	25%	114M	1s
##	33950K	25%	72.0M	1s
##	34000K	25%	108M	1s
##	34050K	25%	107M	1s
##	34100K	25%	116M	1s
##	34150K	25%	82.0M	1s
##	34200K	25%	111M	1s
##	34250K	25%	101M	1s
##	34300K	25%	112M	1s
##	34350K	25%	61.8M	1s
##	34400K	25%	114M	1s
##	34450K	25%	78.2M	1s
##	34500K	25%	108M	1s
##	34550K	25%	95.4M	1s
##	34600K	25%	112M	1s
##	34650K	25%	90.9M	1s
##	34700K	25%	105M	1s
##	34750K	25%	91.0M	1s
##	34800K	25%	94.9M	1s
##	34850K	26%	106M	1s
##	34900K	26%	80.9M	1s
##	34950K	26%	104M	1s
##	35000K	26%	129M	1s
##	35050K	26%	113M	1s
##	35100K	26%	106M	1s
##	35150K	26%	99.4M	1s
##	35200K	26%	124M	1s
##	35250K	26%	97.6M	1s
##	35300K	26%	128M	1s
##	35350K	26%	70.5M	1s
##	35400K	26%	112M	1s
##	35450K	26%	105M	1s
##	35500K	26%	105M	1s
##	35550K	26%	78.8M	1s
##	35600K	26%	100M	1s
##	35650K	26%	124M	1s
##	35700K	26%	108M	1s
##	35750K	26%	105M	1s
##	35800K	26%	110M	1s
##	35850K	26%	143M	1s
##	35900K	26%	31.5M	1s
##	35950K	26%	47.9M	1s
##	36000K	26%	107M	1s
##	36050K	26%	62.5M	1s
##	36100K	26%	78.7M	1s
##	36150K	27%	67.5M	1s
##	36200K	27%	25.8M	1s

##	36250K	27%	101M	1s
##	36300K	27%	80.4M	1s
##	36350K	27%	85.7M	1s
##	36400K	27%	122M	1s
##	36450K	27%	108M	1s
##	36500K	27%	114M	1s
##	36550K	27%	95.3M	1s
##	36600K	27%	117M	1s
##	36650K	27%	122M	1s
##	36700K	27%	120M	1s
##	36750K	27%	72.7M	1s
##	36800K	27%	107M	1s
##	36850K	27%	115M	1s
##	36900K	27%	113M	1s
##	36950K	27%	95.9M	1s
##	37000K	27%	124M	1s
##	37050K	27%	109M	1s
##	37100K	27%	78.7M	1s
##	37150K	27%	81.3M	1s
##	37200K	27%	111M	1s
##	37250K	27%	117M	1s
##	37300K	27%	119M	1s
##	37350K	27%	98.8M	1s
##	37400K	27%	91.6M	1s
##	37450K	27%	80.3M	1s
##	37500K	28%	117M	1s
##	37550K	28%	93.8M	1s
##	37600K	28%	127M	1s
##	37650K	28%	119M	1s
##	37700K	28%	102M	1s
##	37750K	28%	97.2M	1s
##	37800K	28%	119M	1s
##	37850K	28%	115M	1s
##	37900K	28%	108M	1s
##	37950K	28%	86.9M	1s
##	38000K	28%	82.7M	1s
##	38050K	28%	131M	1s
##	38100K	28%	108M	1s
##	38150K	28%	95.4M	1s
##	38200K	28%	107M	1s
##	38250K	28%	114M	1s
##	38300K	28%	82.5M	1s
##	38350K	28%	93.8M	1s
##	38400K	28%	84.2M	1s
##	38450K	28%	117M	1s
##	38500K	28%	104M	1s
##	38550K	28%	99.8M	1s
##	38600K	28%	112M	1s
##	38650K	28%	111M	1s
##	38700K	28%	113M	1s
##	38750K	28%	93.4M	1s
##	38800K	28%	105M	1s
##	38850K	29%	122M	1s
##	38900K	29%	114M	1s

##	38950K	29%	87.6M	1s
##	39000K	29%	104M	1s
##	39050K	29%	131M	1s
##	39100K	29%	109M	1s
##	39150K	29%	85.6M	1s
##	39200K	29%	112M	1s
##	39250K	29%	103M	1s
##	39300K	29%	113M	1s
##	39350K	29%	121M	1s
##	39400K	29%	105M	1s
##	39450K	29%	78.8M	1s
##	39500K	29%	113M	1s
##	39550K	29%	100M	1s
##	39600K	29%	101M	1s
##	39650K	29%	109M	1s
##	39700K	29%	113M	1s
##	39750K	29%	92.1M	1s
##	39800K	29%	120M	1s
##	39850K	29%	100M	1s
##	39900K	29%	114M	1s
##	39950K	29%	105M	1s
##	40000K	29%	110M	1s
##	40050K	29%	104M	1s
##	40100K	29%	115M	1s
##	40150K	29%	110M	1s
##	40200K	30%	112M	1s
##	40250K	30%	129M	1s
##	40300K	30%	106M	1s
##	40350K	30%	84.2M	1s
##	40400K	30%	106M	1s
##	40450K	30%	107M	1s
##	40500K	30%	99.4M	1s
##	40550K	30%	91.4M	1s
##	40600K	30%	111M	1s
##	40650K	30%	103M	1s
##	40700K	30%	121M	1s
##	40750K	30%	88.9M	1s
##	40800K	30%	126M	1s
##	40850K	30%	115M	1s
##	40900K	30%	106M	1s
##	40950K	30%	92.4M	1s
##	41000K	30%	116M	1s
##	41050K	30%	121M	1s
##	41100K	30%	114M	1s
##	41150K	30%	88.9M	1s
##	41200K	30%	116M	1s
##	41250K	30%	103M	1s
##	41300K	30%	105M	1s
##	41350K	30%	102M	1s
##	41400K	30%	111M	1s
##	41450K	30%	121M	1s
##	41500K	30%	121M	1s
##	41550K	31%	86.0M	1s
##	41600K	31%	114M	1s

##	41650K	31%	124M	1s
##	41700K	31%	113M	1s
##	41750K	31%	102M	1s
##	41800K	31%	112M	1s
##	41850K	31%	99.0M	1s
##	41900K	31%	116M	1s
##	41950K	31%	108M	1s
##	42000K	31%	119M	1s
##	42050K	31%	113M	1s
##	42100K	31%	133M	1s
##	42150K	31%	95.2M	1s
##	42200K	31%	101M	1s
##	42250K	31%	114M	1s
##	42300K	31%	114M	1s
##	42350K	31%	91.4M	1s
##	42400K	31%	92.9M	1s
##	42450K	31%	71.6M	1s
##	42500K	31%	73.9M	1s
##	42550K	31%	51.0M	1s
##	42600K	31%	86.4M	1s
##	42650K	31%	46.9M	1s
##	42700K	31%	78.4M	1s
##	42750K	31%	54.4M	1s
##	42800K	31%	57.5M	1s
##	42850K	31%	71.4M	1s
##	42900K	32%	61.0M	1s
##	42950K	32%	99.6M	1s
##	43000K	32%	68.9M	1s
##	43050K	32%	40.8M	1s
##	43100K	32%	138M	1s
##	43150K	32%	36.6M	1s
##	43200K	32%	50.8M	1s
##	43250K	32%	36.2M	1s
##	43300K	32%	66.1M	1s
##	43350K	32%	57.8M	1s
##	43400K	32%	84.0M	1s
##	43450K	32%	41.4M	1s
##	43500K	32%	125M	1s
##	43550K	32%	29.8M	1s
##	43600K	32%	46.0M	1s
##	43650K	32%	48.2M	1s
##	43700K	32%	70.7M	1s
##	43750K	32%	35.0M	1s
##	43800K	32%	92.4M	1s
##	43850K	32%	111M	1s
##	43900K	32%	32.4M	1s
##	43950K	32%	53.4M	1s
##	44000K	32%	145M	1s
##	44050K	32%	61.7M	1s
##	44100K	32%	54.3M	1s
##	44150K	32%	61.3M	1s
##	44200K	33%	141M	1s
##	44250K	33%	137M	1s
##	44300K	33%	120M	1s

##	44350K	33%	64.0M	1s
##	44400K	33%	110M	1s
##	44450K	33%	68.2M	1s
##	44500K	33%	140M	1s
##	44550K	33%	70.0M	1s
##	44600K	33%	147M	1s
##	44650K	33%	130M	1s
##	44700K	33%	33.3M	1s
##	44750K	33%	102M	1s
##	44800K	33%	129M	1s
##	44850K	33%	146M	1s
##	44900K	33%	27.8M	1s
##	44950K	33%	118M	1s
##	45000K	33%	126M	1s
##	45050K	33%	147M	1s
##	45100K	33%	137M	1s
##	45150K	33%	102M	1s
##	45200K	33%	71.9M	1s
##	45250K	33%	62.6M	1s
##	45300K	33%	107M	1s
##	45350K	33%	79.2M	1s
##	45400K	33%	44.9M	1s
##	45450K	33%	66.4M	1s
##	45500K	33%	81.1M	1s
##	45550K	34%	71.2M	1s
##	45600K	34%	95.9M	1s
##	45650K	34%	97.4M	1s
##	45700K	34%	80.0M	1s
##	45750K	34%	61.7M	1s
##	45800K	34%	66.9M	1s
##	45850K	34%	49.0M	1s
##	45900K	34%	35.1M	1s
##	45950K	34%	52.8M	1s
##	46000K	34%	33.4M	1s
##	46050K	34%	92.5M	1s
##	46100K	34%	47.3M	1s
##	46150K	34%	61.3M	1s
##	46200K	34%	60.4M	1s
##	46250K	34%	47.1M	1s
##	46300K	34%	61.7M	1s
##	46350K	34%	40.1M	1s
##	46400K	34%	98.2M	1s
##	46450K	34%	47.0M	1s
##	46500K	34%	38.1M	1s
##	46550K	34%	46.9M	1s
##	46600K	34%	78.1M	1s
##	46650K	34%	120M	1s
##	46700K	34%	33.8M	1s
##	46750K	34%	53.7M	1s
##	46800K	34%	52.9M	1s
##	46850K	34%	67.5M	1s
##	46900K	35%	36.4M	1s
##	46950K	35%	43.0M	1s
##	47000K	35%	69.6M	1s

##	47050K	35%	53.7M	1s
##	47100K	35%	42.0M	1s
##	47150K	35%	37.1M	1s
##	47200K	35%	114M	1s
##	47250K	35%	38.8M	1s
##	47300K	35%	40.4M	1s
##	47350K	35%	62.5M	1s
##	47400K	35%	56.3M	1s
##	47450K	35%	127M	1s
##	47500K	35%	42.7M	1s
##	47550K	35%	40.3M	1s
##	47600K	35%	58.8M	1s
##	47650K	35%	67.9M	1s
##	47700K	35%	107M	1s
##	47750K	35%	25.1M	1s
##	47800K	35%	129M	1s
##	47850K	35%	83.0M	1s
##	47900K	35%	51.9M	1s
##	47950K	35%	56.2M	1s
##	48000K	35%	36.0M	1s
##	48050K	35%	94.6M	1s
##	48100K	35%	104M	1s
##	48150K	35%	44.3M	1s
##	48200K	35%	45.5M	1s
##	48250K	36%	65.6M	1s
##	48300K	36%	121M	1s
##	48350K	36%	63.6M	1s
##	48400K	36%	32.2M	1s
##	48450K	36%	64.3M	1s
##	48500K	36%	76.0M	1s
##	48550K	36%	62.4M	1s
##	48600K	36%	39.3M	1s
##	48650K	36%	57.1M	1s
##	48700K	36%	88.1M	1s
##	48750K	36%	79.2M	1s
##	48800K	36%	54.8M	1s
##	48850K	36%	33.9M	1s
##	48900K	36%	72.2M	1s
##	48950K	36%	75.3M	1s
##	49000K	36%	78.0M	1s
##	49050K	36%	56.6M	1s
##	49100K	36%	52.9M	1s
##	49150K	36%	46.4M	1s
##	49200K	36%	63.9M	1s
##	49250K	36%	94.5M	1s
##	49300K	36%	59.8M	1s
##	49350K	36%	42.1M	1s
##	49400K	36%	58.6M	1s
##	49450K	36%	63.5M	1s
##	49500K	36%	59.9M	1s
##	49550K	36%	33.1M	1s
##	49600K	37%	75.3M	1s
##	49650K	37%	41.2M	1s
##	49700K	37%	52.0M	1s

##	49750K	37%	39.9M	1s
##	49800K	37%	46.7M	1s
##	49850K	37%	58.9M	1s
##	49900K	37%	61.8M	1s
##	49950K	37%	39.0M	1s
##	50000K	37%	60.7M	1s
##	50050K	37%	126M	1s
##	50100K	37%	86.6M	1s
##	50150K	37%	111M	1s
##	50200K	37%	75.7M	1s
##	50250K	37%	121M	1s
##	50300K	37%	96.5M	1s
##	50350K	37%	85.9M	1s
##	50400K	37%	107M	1s
##	50450K	37%	98.8M	1s
##	50500K	37%	93.9M	1s
##	50550K	37%	120M	1s
##	50600K	37%	131M	1s
##	50650K	37%	108M	1s
##	50700K	37%	140M	1s
##	50750K	37%	116M	1s
##	50800K	37%	82.5M	1s
##	50850K	37%	73.6M	1s
##	50900K	38%	135M	1s
##	50950K	38%	126M	1s
##	51000K	38%	147M	1s
##	51050K	38%	133M	1s
##	51100K	38%	139M	1s
##	51150K	38%	55.4M	1s
##	51200K	38%	118M	1s
##	51250K	38%	136M	1s
##	51300K	38%	148M	1s
##	51350K	38%	66.7M	1s
##	51400K	38%	105M	1s
##	51450K	38%	76.9M	1s
##	51500K	38%	73.1M	1s
##	51550K	38%	92.3M	1s
##	51600K	38%	139M	1s
##	51650K	38%	137M	1s
##	51700K	38%	137M	1s
##	51750K	38%	77.0M	1s
##	51800K	38%	89.6M	1s
##	51850K	38%	62.0M	1s
##	51900K	38%	104M	1s
##	51950K	38%	62.3M	1s
##	52000K	38%	36.2M	1s
##	52050K	38%	132M	1s
##	52100K	38%	140M	1s
##	52150K	38%	115M	1s
##	52200K	38%	137M	1s
##	52250K	39%	152M	1s
##	52300K	39%	93.8M	1s
##	52350K	39%	96.7M	1s
##	52400K	39%	128M	1s

##	52450K	39%	75.3M	1s
##	52500K	39%	130M	1s
##	52550K	39%	56.8M	1s
##	52600K	39%	122M	1s
##	52650K	39%	152M	1s
##	52700K	39%	48.2M	1s
##	52750K	39%	121M	1s
##	52800K	39%	48.9M	1s
##	52850K	39%	90.9M	1s
##	52900K	39%	33.0M	1s
##	52950K	39%	31.9M	1s
##	53000K	39%	83.6M	1s
##	53050K	39%	86.7M	1s
##	53100K	39%	21.9M	1s
##	53150K	39%	74.9M	1s
##	53200K	39%	18.5M	1s
##	53250K	39%	89.0M	1s
##	53300K	39%	44.1M	1s
##	53350K	39%	41.3M	1s
##	53400K	39%	134M	1s
##	53450K	39%	139M	1s
##	53500K	39%	151M	1s
##	53550K	39%	60.7M	1s
##	53600K	40%	125M	1s
##	53650K	40%	145M	1s
##	53700K	40%	65.7M	1s
##	53750K	40%	31.9M	1s
##	53800K	40%	56.4M	1s
##	53850K	40%	47.0M	1s
##	53900K	40%	68.9M	1s
##	53950K	40%	27.8M	1s
##	54000K	40%	46.9M	1s
##	54050K	40%	53.1M	1s
##	54100K	40%	55.9M	1s
##	54150K	40%	58.2M	1s
##	54200K	40%	45.3M	1s
##	54250K	40%	41.4M	1s
##	54300K	40%	69.9M	1s
##	54350K	40%	65.7M	1s
##	54400K	40%	47.1M	1s
##	54450K	40%	37.2M	1s
##	54500K	40%	72.0M	1s
##	54550K	40%	51.7M	1s
##	54600K	40%	37.4M	1s
##	54650K	40%	64.4M	1s
##	54700K	40%	59.5M	1s
##	54750K	40%	64.6M	1s
##	54800K	40%	57.9M	1s
##	54850K	40%	58.3M	1s
##	54900K	40%	84.4M	1s
##	54950K	41%	70.2M	1s
##	55000K	41%	84.9M	1s
##	55050K	41%	67.5M	1s
##	55100K	41%	63.1M	1s

##	55150K	41%	38.5M	1s
##	55200K	41%	50.1M	1s
##	55250K	41%	77.7M	1s
##	55300K	41%	77.1M	1s
##	55350K	41%	74.3M	1s
##	55400K	41%	85.6M	1s
##	55450K	41%	36.1M	1s
##	55500K	41%	6.30M	1s
##	55550K	41%	45.6M	1s
##	55600K	41%	65.6M	1s
##	55650K	41%	81.7M	1s
##	55700K	41%	69.8M	1s
##	55750K	41%	63.0M	1s
##	55800K	41%	79.1M	1s
##	55850K	41%	64.7M	1s
##	55900K	41%	96.1M	1s
##	55950K	41%	79.8M	1s
##	56000K	41%	84.9M	1s
##	56050K	41%	40.2M	1s
##	56100K	41%	69.7M	1s
##	56150K	41%	79.9M	1s
##	56200K	41%	103M	1s
##	56250K	41%	104M	1s
##	56300K	42%	102M	1s
##	56350K	42%	37.4M	1s
##	56400K	42%	90.6M	1s
##	56450K	42%	99.6M	1s
##	56500K	42%	101M	1s
##	56550K	42%	89.4M	1s
##	56600K	42%	97.6M	1s
##	56650K	42%	50.9M	1s
##	56700K	42%	63.0M	1s
##	56750K	42%	85.0M	1s
##	56800K	42%	91.6M	1s
##	56850K	42%	66.7M	1s
##	56900K	42%	92.8M	1s
##	56950K	42%	77.6M	1s
##	57000K	42%	38.3M	1s
##	57050K	42%	31.3M	1s
##	57100K	42%	111M	1s
##	57150K	42%	88.1M	1s
##	57200K	42%	104M	1s
##	57250K	42%	76.8M	1s
##	57300K	42%	32.3M	1s
##	57350K	42%	47.2M	1s
##	57400K	42%	98.7M	1s
##	57450K	42%	104M	1s
##	57500K	42%	108M	1s
##	57550K	42%	58.1M	1s
##	57600K	43%	85.9M	1s
##	57650K	43%	106M	1s
##	57700K	43%	115M	1s
##	57750K	43%	86.8M	1s
##	57800K	43%	48.0M	1s

##	57850K	43%	75.0M	1s
##	57900K	43%	54.0M	1s
##	57950K	43%	40.7M	1s
##	58000K	43%	70.3M	1s
##	58050K	43%	110M	1s
##	58100K	43%	30.2M	1s
##	58150K	43%	36.4M	1s
##	58200K	43%	50.3M	1s
##	58250K	43%	48.1M	1s
##	58300K	43%	49.2M	1s
##	58350K	43%	33.9M	1s
##	58400K	43%	113M	1s
##	58450K	43%	54.8M	1s
##	58500K	43%	54.4M	1s
##	58550K	43%	51.2M	1s
##	58600K	43%	73.3M	1s
##	58650K	43%	91.4M	1s
##	58700K	43%	58.9M	1s
##	58750K	43%	58.0M	1s
##	58800K	43%	123M	1s
##	58850K	43%	118M	1s
##	58900K	43%	72.8M	1s
##	58950K	44%	79.2M	1s
##	59000K	44%	46.2M	1s
##	59050K	44%	101M	1s
##	59100K	44%	115M	1s
##	59150K	44%	105M	1s
##	59200K	44%	135M	1s
##	59250K	44%	94.6M	1s
##	59300K	44%	74.2M	1s
##	59350K	44%	87.8M	1s
##	59400K	44%	43.8M	1s
##	59450K	44%	131M	1s
##	59500K	44%	108M	1s
##	59550K	44%	88.0M	1s
##	59600K	44%	49.4M	1s
##	59650K	44%	98.5M	1s
##	59700K	44%	115M	1s
##	59750K	44%	29.9M	1s
##	59800K	44%	40.7M	1s
##	59850K	44%	135M	1s
##	59900K	44%	128M	1s
##	59950K	44%	113M	1s
##	60000K	44%	116M	1s
##	60050K	44%	118M	1s
##	60100K	44%	137M	1s
##	60150K	44%	108M	1s
##	60200K	44%	123M	1s
##	60250K	44%	131M	1s
##	60300K	45%	135M	1s
##	60350K	45%	65.8M	1s
##	60400K	45%	93.1M	1s
##	60450K	45%	136M	1s
##	60500K	45%	99.6M	1s

##	60550K	45%	120M	1s
##	60600K	45%	135M	1s
##	60650K	45%	136M	1s
##	60700K	45%	141M	1s
##	60750K	45%	109M	1s
##	60800K	45%	141M	1s
##	60850K	45%	137M	1s
##	60900K	45%	121M	1s
##	60950K	45%	87.3M	1s
##	61000K	45%	132M	1s
##	61050K	45%	117M	1s
##	61100K	45%	76.4M	1s
##	61150K	45%	100M	1s
##	61200K	45%	76.8M	1s
##	61250K	45%	20.6M	1s
##	61300K	45%	128M	1s
##	61350K	45%	124M	1s
##	61400K	45%	137M	1s
##	61450K	45%	145M	1s
##	61500K	45%	137M	1s
##	61550K	45%	120M	1s
##	61600K	45%	48.0M	1s
##	61650K	46%	85.4M	1s
##	61700K	46%	132M	1s
##	61750K	46%	92.4M	1s
##	61800K	46%	45.3M	1s
##	61850K	46%	53.7M	1s
##	61900K	46%	92.6M	1s
##	61950K	46%	114M	1s
##	62000K	46%	145M	1s
##	62050K	46%	111M	1s
##	62100K	46%	95.6M	1s
##	62150K	46%	65.7M	1s
##	62200K	46%	127M	1s
##	62250K	46%	52.7M	1s
##	62300K	46%	89.6M	1s
##	62350K	46%	56.7M	1s
##	62400K	46%	116M	1s
##	62450K	46%	86.4M	1s
##	62500K	46%	60.5M	1s
##	62550K	46%	44.6M	1s
##	62600K	46%	142M	1s
##	62650K	46%	142M	1s
##	62700K	46%	144M	1s
##	62750K	46%	49.5M	1s
##	62800K	46%	77.9M	1s
##	62850K	46%	141M	1s
##	62900K	46%	84.5M	1s
##	62950K	46%	54.2M	1s
##	63000K	47%	80.7M	1s
##	63050K	47%	144M	1s
##	63100K	47%	149M	1s
##	63150K	47%	51.0M	1s
##	63200K	47%	67.9M	1s

##	63250K	47%	79.1M	1s
##	63300K	47%	149M	1s
##	63350K	47%	65.0M	1s
##	63400K	47%	63.5M	1s
##	63450K	47%	90.4M	1s
##	63500K	47%	103M	1s
##	63550K	47%	52.0M	1s
##	63600K	47%	65.3M	1s
##	63650K	47%	140M	1s
##	63700K	47%	147M	1s
##	63750K	47%	49.3M	1s
##	63800K	47%	50.6M	1s
##	63850K	47%	99.8M	1s
##	63900K	47%	79.7M	1s
##	63950K	47%	67.7M	1s
##	64000K	47%	71.9M	1s
##	64050K	47%	143M	1s
##	64100K	47%	56.9M	1s
##	64150K	47%	63.3M	1s
##	64200K	47%	130M	1s
##	64250K	47%	144M	1s
##	64300K	47%	77.6M	1s
##	64350K	48%	51.9M	1s
##	64400K	48%	101M	1s
##	64450K	48%	77.3M	1s
##	64500K	48%	86.5M	1s
##	64550K	48%	51.5M	1s
##	64600K	48%	84.2M	1s
##	64650K	48%	148M	1s
##	64700K	48%	93.2M	1s
##	64750K	48%	50.8M	1s
##	64800K	48%	84.7M	1s
##	64850K	48%	60.0M	1s
##	64900K	48%	143M	1s
##	64950K	48%	52.0M	1s
##	65000K	48%	136M	1s
##	65050K	48%	145M	1s
##	65100K	48%	33.7M	1s
##	65150K	48%	124M	1s
##	65200K	48%	149M	1s
##	65250K	48%	139M	1s
##	65300K	48%	150M	1s
##	65350K	48%	71.4M	1s
##	65400K	48%	112M	1s
##	65450K	48%	110M	1s
##	65500K	48%	69.1M	1s
##	65550K	48%	45.7M	1s
##	65600K	48%	118M	1s
##	65650K	49%	125M	1s
##	65700K	49%	123M	1s
##	65750K	49%	90.5M	1s
##	65800K	49%	74.4M	1s
##	65850K	49%	104M	1s
##	65900K	49%	149M	1s

##	65950K	49%	95.9M	1s
##	66000K	49%	149M	1s
##	66050K	49%	70.2M	1s
##	66100K	49%	140M	1s
##	66150K	49%	127M	1s
##	66200K	49%	26.8M	1s
##	66250K	49%	148M	1s
##	66300K	49%	150M	1s
##	66350K	49%	118M	1s
##	66400K	49%	151M	1s
##	66450K	49%	146M	1s
##	66500K	49%	141M	1s
##	66550K	49%	130M	1s
##	66600K	49%	38.8M	1s
##	66650K	49%	123M	1s
##	66700K	49%	114M	1s
##	66750K	49%	91.3M	1s
##	66800K	49%	124M	1s
##	66850K	49%	142M	1s
##	66900K	49%	127M	1s
##	66950K	49%	124M	1s
##	67000K	50%	115M	1s
##	67050K	50%	74.2M	1s
##	67100K	50%	136M	1s
##	67150K	50%	126M	1s
##	67200K	50%	146M	1s
##	67250K	50%	110M	1s
##	67300K	50%	71.6M	1s
##	67350K	50%	124M	1s
##	67400K	50%	85.2M	1s
##	67450K	50%	129M	1s
##	67500K	50%	59.3M	1s
##	67550K	50%	127M	1s
##	67600K	50%	29.4M	1s
##	67650K	50%	147M	1s
##	67700K	50%	141M	1s
##	67750K	50%	124M	1s
##	67800K	50%	132M	1s
##	67850K	50%	63.0M	1s
##	67900K	50%	89.8M	1s
##	67950K	50%	78.7M	1s
##	68000K	50%	152M	1s
##	68050K	50%	99.6M	1s
##	68100K	50%	121M	1s
##	68150K	50%	51.4M	1s
##	68200K	50%	122M	1s
##	68250K	50%	71.4M	1s
##	68300K	50%	77.8M	1s
##	68350K	51%	52.8M	1s
##	68400K	51%	83.7M	1s
##	68450K	51%	115M	1s
##	68500K	51%	114M	1s
##	68550K	51%	88.5M	1s
##	68600K	51%	107M	1s

##	68650K	51%	143M	1s
##	68700K	51%	52.9M	1s
##	68750K	51%	28.3M	1s
##	68800K	51%	93.7M	1s
##	68850K	51%	48.2M	1s
##	68900K	51%	107M	1s
##	68950K	51%	90.1M	1s
##	69000K	51%	88.0M	1s
##	69050K	51%	143M	1s
##	69100K	51%	45.8M	1s
##	69150K	51%	81.3M	1s
##	69200K	51%	74.5M	1s
##	69250K	51%	98.1M	1s
##	69300K	51%	86.3M	1s
##	69350K	51%	100M	1s
##	69400K	51%	76.5M	1s
##	69450K	51%	46.7M	1s
##	69500K	51%	60.6M	1s
##	69550K	51%	57.7M	1s
##	69600K	51%	140M	1s
##	69650K	51%	54.6M	1s
##	69700K	52%	61.0M	1s
##	69750K	52%	57.0M	1s
##	69800K	52%	41.2M	1s
##	69850K	52%	104M	1s
##	69900K	52%	62.0M	1s
##	69950K	52%	110M	1s
##	70000K	52%	117M	1s
##	70050K	52%	42.7M	1s
##	70100K	52%	55.9M	1s
##	70150K	52%	68.4M	1s
##	70200K	52%	78.7M	1s
##	70250K	52%	53.6M	1s
##	70300K	52%	57.3M	1s
##	70350K	52%	119M	1s
##	70400K	52%	68.7M	1s
##	70450K	52%	84.6M	1s
##	70500K	52%	46.9M	1s
##	70550K	52%	118M	1s
##	70600K	52%	43.0M	1s
##	70650K	52%	78.0M	1s
##	70700K	52%	60.8M	1s
##	70750K	52%	117M	1s
##	70800K	52%	61.1M	1s
##	70850K	52%	52.6M	1s
##	70900K	52%	97.3M	1s
##	70950K	52%	122M	1s
##	71000K	52%	147M	1s
##	71050K	53%	113M	1s
##	71100K	53%	54.3M	1s
##	71150K	53%	28.7M	1s
##	71200K	53%	124M	1s
##	71250K	53%	26.2M	1s
##	71300K	53%	135M	1s

##	71350K	53%	126M	1s
##	71400K	53%	145M	1s
##	71450K	53%	140M	1s
##	71500K	53%	141M	1s
##	71550K	53%	121M	1s
##	71600K	53%	148M	1s
##	71650K	53%	97.9M	1s
##	71700K	53%	139M	1s
##	71750K	53%	124M	1s
##	71800K	53%	130M	1s
##	71850K	53%	60.3M	1s
##	71900K	53%	143M	1s
##	71950K	53%	49.5M	1s
##	72000K	53%	104M	1s
##	72050K	53%	80.7M	1s
##	72100K	53%	31.4M	1s
##	72150K	53%	124M	1s
##	72200K	53%	149M	1s
##	72250K	53%	139M	1s
##	72300K	53%	140M	1s
##	72350K	54%	125M	1s
##	72400K	54%	127M	1s
##	72450K	54%	150M	1s
##	72500K	54%	144M	1s
##	72550K	54%	124M	1s
##	72600K	54%	143M	1s
##	72650K	54%	73.1M	1s
##	72700K	54%	67.4M	1s
##	72750K	54%	115M	1s
##	72800K	54%	118M	1s
##	72850K	54%	84.4M	1s
##	72900K	54%	109M	1s
##	72950K	54%	95.1M	1s
##	73000K	54%	147M	1s
##	73050K	54%	132M	1s
##	73100K	54%	148M	1s
##	73150K	54%	124M	1s
##	73200K	54%	138M	1s
##	73250K	54%	150M	1s
##	73300K	54%	146M	1s
##	73350K	54%	126M	1s
##	73400K	54%	142M	1s
##	73450K	54%	143M	1s
##	73500K	54%	141M	1s
##	73550K	54%	48.8M	1s
##	73600K	54%	140M	1s
##	73650K	54%	18.1M	1s
##	73700K	55%	23.4M	1s
##	73750K	55%	97.2M	1s
##	73800K	55%	106M	1s
##	73850K	55%	107M	1s
##	73900K	55%	49.3M	1s
##	73950K	55%	18.2M	1s
##	74000K	55%	103M	1s

##	74050K	55%	103M	1s
##	74100K	55%	116M	1s
##	74150K	55%	92.2M	1s
##	74200K	55%	114M	1s
##	74250K	55%	53.6M	1s
##	74300K	55%	92.7M	1s
##	74350K	55%	38.0M	1s
##	74400K	55%	68.7M	1s
##	74450K	55%	59.0M	1s
##	74500K	55%	61.2M	1s
##	74550K	55%	69.8M	1s
##	74600K	55%	119M	1s
##	74650K	55%	104M	1s
##	74700K	55%	56.1M	1s
##	74750K	55%	41.4M	1s
##	74800K	55%	63.8M	1s
##	74850K	55%	45.3M	1s
##	74900K	55%	52.1M	1s
##	74950K	55%	55.9M	1s
##	75000K	55%	57.3M	1s
##	75050K	56%	60.4M	1s
##	75100K	56%	59.0M	1s
##	75150K	56%	59.0M	1s
##	75200K	56%	72.1M	1s
##	75250K	56%	72.3M	1s
##	75300K	56%	62.7M	1s
##	75350K	56%	48.8M	1s
##	75400K	56%	59.2M	1s
##	75450K	56%	65.2M	1s
##	75500K	56%	72.3M	1s
##	75550K	56%	45.2M	1s
##	75600K	56%	57.7M	1s
##	75650K	56%	79.7M	1s
##	75700K	56%	84.9M	1s
##	75750K	56%	67.8M	1s
##	75800K	56%	63.9M	1s
##	75850K	56%	72.0M	1s
##	75900K	56%	79.4M	1s
##	75950K	56%	68.8M	1s
##	76000K	56%	62.1M	1s
##	76050K	56%	70.5M	1s
##	76100K	56%	69.7M	1s
##	76150K	56%	61.9M	1s
##	76200K	56%	74.3M	1s
##	76250K	56%	91.9M	1s
##	76300K	56%	77.5M	1s
##	76350K	56%	62.2M	1s
##	76400K	57%	92.4M	1s
##	76450K	57%	69.1M	1s
##	76500K	57%	78.7M	1s
##	76550K	57%	68.2M	1s
##	76600K	57%	75.9M	1s
##	76650K	57%	41.5M	1s
##	76700K	57%	77.2M	1s

##	76750K	57%	76.9M	1s
##	76800K	57%	88.2M	1s
##	76850K	57%	97.2M	1s
##	76900K	57%	97.8M	1s
##	76950K	57%	80.0M	1s
##	77000K	57%	87.1M	1s
##	77050K	57%	85.2M	1s
##	77100K	57%	82.7M	1s
##	77150K	57%	83.0M	1s
##	77200K	57%	101M	1s
##	77250K	57%	93.8M	1s
##	77300K	57%	85.4M	1s
##	77350K	57%	84.8M	1s
##	77400K	57%	73.8M	1s
##	77450K	57%	66.3M	1s
##	77500K	57%	72.2M	1s
##	77550K	57%	82.5M	1s
##	77600K	57%	87.2M	1s
##	77650K	57%	93.4M	1s
##	77700K	57%	103M	1s
##	77750K	58%	65.1M	1s
##	77800K	58%	95.6M	1s
##	77850K	58%	86.2M	1s
##	77900K	58%	83.7M	1s
##	77950K	58%	85.6M	1s
##	78000K	58%	99.3M	1s
##	78050K	58%	47.7M	1s
##	78100K	58%	93.2M	1s
##	78150K	58%	92.1M	1s
##	78200K	58%	102M	1s
##	78250K	58%	101M	1s
##	78300K	58%	71.5M	1s
##	78350K	58%	80.4M	1s
##	78400K	58%	98.9M	1s
##	78450K	58%	103M	1s
##	78500K	58%	97.0M	1s
##	78550K	58%	71.9M	1s
##	78600K	58%	93.6M	1s
##	78650K	58%	109M	1s
##	78700K	58%	108M	1s
##	78750K	58%	90.4M	1s
##	78800K	58%	101M	1s
##	78850K	58%	104M	1s
##	78900K	58%	100M	1s
##	78950K	58%	89.4M	1s
##	79000K	58%	113M	1s
##	79050K	59%	93.4M	1s
##	79100K	59%	102M	1s
##	79150K	59%	92.5M	1s
##	79200K	59%	103M	1s
##	79250K	59%	108M	1s
##	79300K	59%	109M	1s
##	79350K	59%	85.4M	1s
##	79400K	59%	24.4M	1s

##	79450K	59%	98.5M	1s
##	79500K	59%	108M	1s
##	79550K	59%	93.2M	1s
##	79600K	59%	76.1M	1s
##	79650K	59%	87.5M	1s
##	79700K	59%	115M	1s
##	79750K	59%	103M	1s
##	79800K	59%	44.2M	1s
##	79850K	59%	111M	1s
##	79900K	59%	117M	1s
##	79950K	59%	88.6M	1s
##	80000K	59%	118M	1s
##	80050K	59%	97.9M	1s
##	80100K	59%	90.8M	1s
##	80150K	59%	95.9M	1s
##	80200K	59%	107M	1s
##	80250K	59%	117M	1s
##	80300K	59%	119M	1s
##	80350K	59%	94.4M	1s
##	80400K	60%	103M	1s
##	80450K	60%	110M	1s
##	80500K	60%	100M	1s
##	80550K	60%	89.8M	1s
##	80600K	60%	119M	1s
##	80650K	60%	115M	1s
##	80700K	60%	101M	1s
##	80750K	60%	82.0M	1s
##	80800K	60%	89.1M	1s
##	80850K	60%	103M	1s
##	80900K	60%	101M	1s
##	80950K	60%	79.3M	1s
##	81000K	60%	102M	1s
##	81050K	60%	73.1M	1s
##	81100K	60%	90.0M	1s
##	81150K	60%	75.3M	1s
##	81200K	60%	87.8M	1s
##	81250K	60%	91.4M	1s
##	81300K	60%	92.0M	1s
##	81350K	60%	101M	1s
##	81400K	60%	84.3M	1s
##	81450K	60%	90.7M	1s
##	81500K	60%	99.4M	1s
##	81550K	60%	84.7M	1s
##	81600K	60%	99.0M	1s
##	81650K	60%	108M	1s
##	81700K	60%	91.6M	1s
##	81750K	61%	87.5M	1s
##	81800K	61%	110M	1s
##	81850K	61%	114M	1s
##	81900K	61%	103M	1s
##	81950K	61%	87.2M	1s
##	82000K	61%	104M	1s
##	82050K	61%	104M	1s
##	82100K	61%	92.8M	1s

##	82150K	61%	91.5M	1s
##	82200K	61%	92.5M	1s
##	82250K	61%	105M	1s
##	82300K	61%	112M	1s
##	82350K	61%	86.0M	1s
##	82400K	61%	120M	1s
##	82450K	61%	118M	1s
##	82500K	61%	103M	1s
##	82550K	61%	86.4M	1s
##	82600K	61%	89.6M	1s
##	82650K	61%	97.2M	1s
##	82700K	61%	102M	1s
##	82750K	61%	91.9M	1s
##	82800K	61%	95.9M	1s
##	82850K	61%	118M	1s
##	82900K	61%	96.4M	1s
##	82950K	61%	92.3M	1s
##	83000K	61%	101M	1s
##	83050K	61%	56.7M	1s
##	83100K	62%	103M	1s
##	83150K	62%	79.3M	1s
##	83200K	62%	93.9M	1s
##	83250K	62%	135M	1s
##	83300K	62%	127M	1s
##	83350K	62%	88.1M	1s
##	83400K	62%	130M	1s
##	83450K	62%	130M	1s
##	83500K	62%	49.8M	1s
##	83550K	62%	107M	1s
##	83600K	62%	136M	1s
##	83650K	62%	132M	1s
##	83700K	62%	128M	1s
##	83750K	62%	113M	1s
##	83800K	62%	134M	1s
##	83850K	62%	89.4M	1s
##	83900K	62%	85.3M	1s
##	83950K	62%	85.4M	1s
##	84000K	62%	138M	1s
##	84050K	62%	136M	1s
##	84100K	62%	115M	1s
##	84150K	62%	105M	1s
##	84200K	62%	89.8M	1s
##	84250K	62%	113M	1s
##	84300K	62%	120M	1s
##	84350K	62%	98.1M	1s
##	84400K	62%	107M	1s
##	84450K	63%	136M	1s
##	84500K	63%	138M	1s
##	84550K	63%	112M	1s
##	84600K	63%	109M	1s
##	84650K	63%	142M	1s
##	84700K	63%	101M	1s
##	84750K	63%	117M	1s
##	84800K	63%	140M	1s

##	84850K	63%	123M	1s
##	84900K	63%	127M	1s
##	84950K	63%	102M	1s
##	85000K	63%	77.8M	1s
##	85050K	63%	91.0M	1s
##	85100K	63%	105M	1s
##	85150K	63%	97.1M	1s
##	85200K	63%	143M	1s
##	85250K	63%	113M	1s
##	85300K	63%	103M	1s
##	85350K	63%	92.4M	1s
##	85400K	63%	76.7M	1s
##	85450K	63%	142M	1s
##	85500K	63%	113M	1s
##	85550K	63%	105M	1s
##	85600K	63%	135M	1s
##	85650K	63%	119M	1s
##	85700K	63%	131M	1s
##	85750K	63%	96.0M	1s
##	85800K	64%	116M	1s
##	85850K	64%	89.8M	1s
##	85900K	64%	120M	1s
##	85950K	64%	103M	1s
##	86000K	64%	102M	1s
##	86050K	64%	113M	1s
##	86100K	64%	123M	1s
##	86150K	64%	103M	1s
##	86200K	64%	132M	1s
##	86250K	64%	123M	1s
##	86300K	64%	98.8M	1s
##	86350K	64%	96.3M	1s
##	86400K	64%	123M	1s
##	86450K	64%	115M	1s
##	86500K	64%	141M	1s
##	86550K	64%	97.5M	1s
##	86600K	64%	104M	1s
##	86650K	64%	93.4M	1s
##	86700K	64%	130M	1s
##	86750K	64%	104M	1s
##	86800K	64%	115M	1s
##	86850K	64%	85.1M	1s
##	86900K	64%	89.8M	1s
##	86950K	64%	119M	1s
##	87000K	64%	107M	1s
##	87050K	64%	122M	1s
##	87100K	65%	120M	1s
##	87150K	65%	83.0M	1s
##	87200K	65%	106M	1s
##	87250K	65%	124M	1s
##	87300K	65%	110M	1s
##	87350K	65%	112M	1s
##	87400K	65%	100M	1s
##	87450K	65%	118M	1s
##	87500K	65%	127M	1s

##	87550K	65%	102M	1s
##	87600K	65%	109M	1s
##	87650K	65%	92.7M	1s
##	87700K	65%	114M	1s
##	87750K	65%	92.4M	1s
##	87800K	65%	114M	1s
##	87850K	65%	99.2M	1s
##	87900K	65%	142M	1s
##	87950K	65%	93.1M	1s
##	88000K	65%	115M	1s
##	88050K	65%	108M	1s
##	88100K	65%	129M	1s
##	88150K	65%	100M	1s
##	88200K	65%	130M	1s
##	88250K	65%	116M	1s
##	88300K	65%	107M	1s
##	88350K	65%	102M	1s
##	88400K	65%	97.2M	1s
##	88450K	66%	140M	1s
##	88500K	66%	135M	1s
##	88550K	66%	105M	1s
##	88600K	66%	72.6M	1s
##	88650K	66%	98.2M	1s
##	88700K	66%	90.1M	1s
##	88750K	66%	102M	1s
##	88800K	66%	108M	1s
##	88850K	66%	143M	1s
##	88900K	66%	105M	1s
##	88950K	66%	88.6M	1s
##	89000K	66%	108M	1s
##	89050K	66%	105M	1s
##	89100K	66%	98.6M	1s
##	89150K	66%	88.0M	1s
##	89200K	66%	121M	1s
##	89250K	66%	132M	1s
##	89300K	66%	133M	1s
##	89350K	66%	113M	1s
##	89400K	66%	107M	1s
##	89450K	66%	115M	1s
##	89500K	66%	87.6M	1s
##	89550K	66%	86.9M	1s
##	89600K	66%	129M	1s
##	89650K	66%	111M	1s
##	89700K	66%	115M	1s
##	89750K	66%	111M	1s
##	89800K	67%	130M	1s
##	89850K	67%	133M	1s
##	89900K	67%	101M	1s
##	89950K	67%	89.5M	1s
##	90000K	67%	99.1M	1s
##	90050K	67%	111M	1s
##	90100K	67%	129M	1s
##	90150K	67%	101M	1s
##	90200K	67%	115M	1s

##	90250K	67%	104M	1s
##	90300K	67%	102M	1s
##	90350K	67%	76.9M	1s
##	90400K	67%	102M	1s
##	90450K	67%	108M	1s
##	90500K	67%	135M	1s
##	90550K	67%	104M	1s
##	90600K	67%	106M	1s
##	90650K	67%	111M	1s
##	90700K	67%	106M	1s
##	90750K	67%	84.1M	1s
##	90800K	67%	109M	1s
##	90850K	67%	135M	1s
##	90900K	67%	102M	1s
##	90950K	67%	105M	1s
##	91000K	67%	109M	1s
##	91050K	67%	96.6M	1s
##	91100K	67%	83.4M	1s
##	91150K	68%	96.3M	1s
##	91200K	68%	130M	1s
##	91250K	68%	117M	1s
##	91300K	68%	140M	1s
##	91350K	68%	108M	1s
##	91400K	68%	135M	1s
##	91450K	68%	104M	1s
##	91500K	68%	114M	1s
##	91550K	68%	119M	1s
##	91600K	68%	112M	1s
##	91650K	68%	116M	1s
##	91700K	68%	138M	1s
##	91750K	68%	117M	1s
##	91800K	68%	139M	1s
##	91850K	68%	137M	1s
##	91900K	68%	144M	1s
##	91950K	68%	118M	1s
##	92000K	68%	125M	1s
##	92050K	68%	134M	1s
##	92100K	68%	113M	1s
##	92150K	68%	108M	1s
##	92200K	68%	125M	1s
##	92250K	68%	129M	1s
##	92300K	68%	147M	1s
##	92350K	68%	103M	1s
##	92400K	68%	128M	1s
##	92450K	68%	114M	1s
##	92500K	69%	104M	1s
##	92550K	69%	90.0M	1s
##	92600K	69%	109M	0s
##	92650K	69%	127M	0s
##	92700K	69%	128M	0s
##	92750K	69%	115M	0s
##	92800K	69%	122M	0s
##	92850K	69%	100M	0s
##	92900K	69%	119M	0s

##	92950K	69%	124M	0s
##	93000K	69%	125M	0s
##	93050K	69%	139M	0s
##	93100K	69%	144M	0s
##	93150K	69%	92.0M	0s
##	93200K	69%	134M	0s
##	93250K	69%	110M	0s
##	93300K	69%	137M	0s
##	93350K	69%	93.6M	0s
##	93400K	69%	98.3M	0s
##	93450K	69%	142M	0s
##	93500K	69%	126M	0s
##	93550K	69%	90.9M	0s
##	93600K	69%	104M	0s
##	93650K	69%	98.6M	0s
##	93700K	69%	124M	0s
##	93750K	69%	110M	0s
##	93800K	70%	108M	0s
##	93850K	70%	74.7M	0s
##	93900K	70%	146M	0s
##	93950K	70%	112M	0s
##	94000K	70%	121M	0s
##	94050K	70%	128M	0s
##	94100K	70%	137M	0s
##	94150K	70%	84.2M	0s
##	94200K	70%	39.1M	0s
##	94250K	70%	43.1M	0s
##	94300K	70%	110M	0s
##	94350K	70%	43.1M	0s
##	94400K	70%	101M	0s
##	94450K	70%	40.1M	0s
##	94500K	70%	63.1M	0s
##	94550K	70%	69.1M	0s
##	94600K	70%	49.1M	0s
##	94650K	70%	102M	0s
##	94700K	70%	73.9M	0s
##	94750K	70%	37.9M	0s
##	94800K	70%	47.9M	0s
##	94850K	70%	57.0M	0s
##	94900K	70%	72.7M	0s
##	94950K	70%	88.0M	0s
##	95000K	70%	36.6M	0s
##	95050K	70%	96.3M	0s
##	95100K	70%	97.6M	0s
##	95150K	71%	37.7M	0s
##	95200K	71%	96.3M	0s
##	95250K	71%	62.9M	0s
##	95300K	71%	46.8M	0s
##	95350K	71%	75.8M	0s
##	95400K	71%	107M	0s
##	95450K	71%	40.5M	0s
##	95500K	71%	91.2M	0s
##	95550K	71%	33.7M	0s
##	95600K	71%	99.2M	0s

##	95650K	71%	82.9M	0s
##	95700K	71%	50.0M	0s
##	95750K	71%	75.5M	0s
##	95800K	71%	71.5M	0s
##	95850K	71%	60.5M	0s
##	95900K	71%	117M	0s
##	95950K	71%	35.2M	0s
##	96000K	71%	98.3M	0s
##	96050K	71%	79.0M	0s
##	96100K	71%	105M	0s
##	96150K	71%	50.0M	0s
##	96200K	71%	94.9M	0s
##	96250K	71%	52.2M	0s
##	96300K	71%	82.4M	0s
##	96350K	71%	59.0M	0s
##	96400K	71%	61.4M	0s
##	96450K	71%	71.7M	0s
##	96500K	72%	103M	0s
##	96550K	72%	48.7M	0s
##	96600K	72%	79.6M	0s
##	96650K	72%	102M	0s
##	96700K	72%	48.7M	0s
##	96750K	72%	92.7M	0s
##	96800K	72%	64.8M	0s
##	96850K	72%	74.2M	0s
##	96900K	72%	59.6M	0s
##	96950K	72%	86.9M	0s
##	97000K	72%	62.3M	0s
##	97050K	72%	83.0M	0s
##	97100K	72%	70.4M	0s
##	97150K	72%	52.3M	0s
##	97200K	72%	96.0M	0s
##	97250K	72%	7.58M	0s
##	97300K	72%	102M	0s
##	97350K	72%	83.8M	0s
##	97400K	72%	105M	0s
##	97450K	72%	135M	0s
##	97500K	72%	143M	0s
##	97550K	72%	114M	0s
##	97600K	72%	145M	0s
##	97650K	72%	143M	0s
##	97700K	72%	140M	0s
##	97750K	72%	119M	0s
##	97800K	72%	136M	0s
##	97850K	73%	141M	0s
##	97900K	73%	145M	0s
##	97950K	73%	107M	0s
##	98000K	73%	132M	0s
##	98050K	73%	135M	0s
##	98100K	73%	119M	0s
##	98150K	73%	124M	0s
##	98200K	73%	126M	0s
##	98250K	73%	123M	0s
##	98300K	73%	103M	0s

##	98350K	73%	101M	0s
##	98400K	73%	95.6M	0s
##	98450K	73%	137M	0s
##	98500K	73%	140M	0s
##	98550K	73%	46.6M	0s
##	98600K	73%	123M	0s
##	98650K	73%	142M	0s
##	98700K	73%	144M	0s
##	98750K	73%	118M	0s
##	98800K	73%	142M	0s
##	98850K	73%	142M	0s
##	98900K	73%	132M	0s
##	98950K	73%	118M	0s
##	99000K	73%	75.5M	0s
##	99050K	73%	45.9M	0s
##	99100K	73%	40.5M	0s
##	99150K	73%	38.6M	0s
##	99200K	74%	48.8M	0s
##	99250K	74%	23.6M	0s
##	99300K	74%	35.8M	0s
##	99350K	74%	44.9M	0s
##	99400K	74%	47.6M	0s
##	99450K	74%	45.4M	0s
##	99500K	74%	47.1M	0s
##	99550K	74%	75.4M	0s
##	99600K	74%	44.4M	0s
##	99650K	74%	100M	0s
##	99700K	74%	40.0M	0s
##	99750K	74%	86.6M	0s
##	99800K	74%	52.6M	0s
##	99850K	74%	35.0M	0s
##	99900K	74%	43.6M	0s
##	99950K	74%	38.5M	0s
##	100000K	74%	46.1M	0s
##	100050K	74%	56.6M	0s
##	100100K	74%	62.3M	0s
##	100150K	74%	34.8M	0s
##	100200K	74%	102M	0s
##	100250K	74%	68.5M	0s
##	100300K	74%	37.5M	0s
##	100350K	74%	78.6M	0s
##	100400K	74%	37.6M	0s
##	100450K	74%	63.9M	0s
##	100500K	75%	90.9M	0s
##	100550K	75%	31.5M	0s
##	100600K	75%	88.2M	0s
##	100650K	75%	109M	0s
##	100700K	75%	35.1M	0s
##	100750K	75%	62.2M	0s
##	100800K	75%	51.6M	0s
##	100850K	75%	48.0M	0s
##	100900K	75%	108M	0s
##	100950K	75%	50.1M	0s
##	101000K	75%	53.0M	0s

## 101050K	75%	88.4M	0s
## 101100K	75%	71.7M	0s
## 101150K	75%	87.8M	0s
## 101200K	75%	116M	0s
## 101250K	75%	67.4M	0s
## 101300K	75%	97.7M	0s
## 101350K	75%	112M	0s
## 101400K	75%	70.9M	0s
## 101450K	75%	99.7M	0s
## 101500K	75%	111M	0s
## 101550K	75%	95.8M	0s
## 101600K	75%	62.5M	0s
## 101650K	75%	106M	0s
## 101700K	75%	73.5M	0s
## 101750K	75%	80.3M	0s
## 101800K	75%	90.4M	0s
## 101850K	76%	117M	0s
## 101900K	76%	83.2M	0s
## 101950K	76%	87.1M	0s
## 102000K	76%	117M	0s
## 102050K	76%	74.0M	0s
## 102100K	76%	76.1M	0s
## 102150K	76%	90.9M	0s
## 102200K	76%	115M	0s
## 102250K	76%	117M	0s
## 102300K	76%	127M	0s
## 102350K	76%	63.4M	0s
## 102400K	76%	98.2M	0s
## 102450K	76%	103M	0s
## 102500K	76%	75.2M	0s
## 102550K	76%	93.5M	0s
## 102600K	76%	75.8M	0s
## 102650K	76%	105M	0s
## 102700K	76%	113M	0s
## 102750K	76%	54.2M	0s
## 102800K	76%	83.2M	0s
## 102850K	76%	105M	0s
## 102900K	76%	125M	0s
## 102950K	76%	112M	0s
## 103000K	76%	130M	0s
## 103050K	76%	110M	0s
## 103100K	76%	117M	0s
## 103150K	76%	89.0M	0s
## 103200K	77%	108M	0s
## 103250K	77%	93.4M	0s
## 103300K	77%	98.0M	0s
## 103350K	77%	101M	0s
## 103400K	77%	104M	0s
## 103450K	77%	135M	0s
## 103500K	77%	145M	0s
## 103550K	77%	22.4M	0s
## 103600K	77%	40.2M	0s
## 103650K	77%	84.9M	0s
## 103700K	77%	81.5M	0s

## 103750K	77%	94.4M	0s
## 103800K	77%	99.0M	0s
## 103850K	77%	116M	0s
## 103900K	77%	118M	0s
## 103950K	77%	95.2M	0s
## 104000K	77%	143M	0s
## 104050K	77%	52.1M	0s
## 104100K	77%	64.5M	0s
## 104150K	77%	112M	0s
## 104200K	77%	138M	0s
## 104250K	77%	132M	0s
## 104300K	77%	23.2M	0s
## 104350K	77%	91.2M	0s
## 104400K	77%	108M	0s
## 104450K	77%	136M	0s
## 104500K	77%	142M	0s
## 104550K	78%	110M	0s
## 104600K	78%	129M	0s
## 104650K	78%	143M	0s
## 104700K	78%	134M	0s
## 104750K	78%	91.8M	0s
## 104800K	78%	138M	0s
## 104850K	78%	127M	0s
## 104900K	78%	105M	0s
## 104950K	78%	69.9M	0s
## 105000K	78%	54.6M	0s
## 105050K	78%	37.4M	0s
## 105100K	78%	113M	0s
## 105150K	78%	26.6M	0s
## 105200K	78%	92.3M	0s
## 105250K	78%	102M	0s
## 105300K	78%	109M	0s
## 105350K	78%	90.1M	0s
## 105400K	78%	102M	0s
## 105450K	78%	113M	0s
## 105500K	78%	117M	0s
## 105550K	78%	32.8M	0s
## 105600K	78%	53.4M	0s
## 105650K	78%	93.5M	0s
## 105700K	78%	74.3M	0s
## 105750K	78%	86.2M	0s
## 105800K	78%	41.8M	0s
## 105850K	78%	105M	0s
## 105900K	79%	34.7M	0s
## 105950K	79%	75.8M	0s
## 106000K	79%	120M	0s
## 106050K	79%	77.7M	0s
## 106100K	79%	109M	0s
## 106150K	79%	75.4M	0s
## 106200K	79%	71.5M	0s
## 106250K	79%	58.6M	0s
## 106300K	79%	112M	0s
## 106350K	79%	64.9M	0s
## 106400K	79%	58.1M	0s

## 106450K	79%	79.3M	0s
## 106500K	79%	104M	0s
## 106550K	79%	90.5M	0s
## 106600K	79%	41.8M	0s
## 106650K	79%	52.1M	0s
## 106700K	79%	108M	0s
## 106750K	79%	71.7M	0s
## 106800K	79%	108M	0s
## 106850K	79%	83.0M	0s
## 106900K	79%	58.6M	0s
## 106950K	79%	95.6M	0s
## 107000K	79%	101M	0s
## 107050K	79%	81.8M	0s
## 107100K	79%	63.3M	0s
## 107150K	79%	69.7M	0s
## 107200K	79%	95.8M	0s
## 107250K	80%	121M	0s
## 107300K	80%	58.1M	0s
## 107350K	80%	52.8M	0s
## 107400K	80%	77.9M	0s
## 107450K	80%	85.0M	0s
## 107500K	80%	90.8M	0s
## 107550K	80%	73.9M	0s
## 107600K	80%	60.1M	0s
## 107650K	80%	97.4M	0s
## 107700K	80%	97.5M	0s
## 107750K	80%	59.5M	0s
## 107800K	80%	110M	0s
## 107850K	80%	96.8M	0s
## 107900K	80%	131M	0s
## 107950K	80%	53.4M	0s
## 108000K	80%	104M	0s
## 108050K	80%	88.1M	0s
## 108100K	80%	69.7M	0s
## 108150K	80%	94.5M	0s
## 108200K	80%	110M	0s
## 108250K	80%	78.5M	0s
## 108300K	80%	97.8M	0s
## 108350K	80%	47.0M	0s
## 108400K	80%	50.0M	0s
## 108450K	80%	95.3M	0s
## 108500K	80%	131M	0s
## 108550K	81%	103M	0s
## 108600K	81%	106M	0s
## 108650K	81%	108M	0s
## 108700K	81%	111M	0s
## 108750K	81%	52.8M	0s
## 108800K	81%	76.7M	0s
## 108850K	81%	103M	0s
## 108900K	81%	109M	0s
## 108950K	81%	74.6M	0s
## 109000K	81%	116M	0s
## 109050K	81%	73.5M	0s
## 109100K	81%	120M	0s

## 109150K	81%	54.5M	0s
## 109200K	81%	110M	0s
## 109250K	81%	102M	0s
## 109300K	81%	107M	0s
## 109350K	81%	117M	0s
## 109400K	81%	85.1M	0s
## 109450K	81%	101M	0s
## 109500K	81%	39.0M	0s
## 109550K	81%	71.8M	0s
## 109600K	81%	86.6M	0s
## 109650K	81%	109M	0s
## 109700K	81%	127M	0s
## 109750K	81%	98.4M	0s
## 109800K	81%	68.1M	0s
## 109850K	81%	93.3M	0s
## 109900K	82%	112M	0s
## 109950K	82%	91.7M	0s
## 110000K	82%	114M	0s
## 110050K	82%	106M	0s
## 110100K	82%	130M	0s
## 110150K	82%	100M	0s
## 110200K	82%	56.4M	0s
## 110250K	82%	116M	0s
## 110300K	82%	110M	0s
## 110350K	82%	83.4M	0s
## 110400K	82%	115M	0s
## 110450K	82%	111M	0s
## 110500K	82%	56.6M	0s
## 110550K	82%	98.0M	0s
## 110600K	82%	94.7M	0s
## 110650K	82%	87.0M	0s
## 110700K	82%	83.0M	0s
## 110750K	82%	92.3M	0s
## 110800K	82%	112M	0s
## 110850K	82%	126M	0s
## 110900K	82%	76.7M	0s
## 110950K	82%	88.3M	0s
## 111000K	82%	113M	0s
## 111050K	82%	110M	0s
## 111100K	82%	108M	0s
## 111150K	82%	89.2M	0s
## 111200K	82%	105M	0s
## 111250K	83%	115M	0s
## 111300K	83%	105M	0s
## 111350K	83%	80.6M	0s
## 111400K	83%	94.2M	0s
## 111450K	83%	97.4M	0s
## 111500K	83%	116M	0s
## 111550K	83%	99.5M	0s
## 111600K	83%	47.8M	0s
## 111650K	83%	97.4M	0s
## 111700K	83%	115M	0s
## 111750K	83%	94.2M	0s
## 111800K	83%	123M	0s

## 111850K	83%	106M	0s
## 111900K	83%	115M	0s
## 111950K	83%	85.3M	0s
## 112000K	83%	137M	0s
## 112050K	83%	116M	0s
## 112100K	83%	102M	0s
## 112150K	83%	98.1M	0s
## 112200K	83%	96.8M	0s
## 112250K	83%	134M	0s
## 112300K	83%	57.9M	0s
## 112350K	83%	86.6M	0s
## 112400K	83%	98.0M	0s
## 112450K	83%	112M	0s
## 112500K	83%	137M	0s
## 112550K	83%	96.9M	0s
## 112600K	84%	117M	0s
## 112650K	84%	111M	0s
## 112700K	84%	123M	0s
## 112750K	84%	73.6M	0s
## 112800K	84%	100M	0s
## 112850K	84%	102M	0s
## 112900K	84%	106M	0s
## 112950K	84%	110M	0s
## 113000K	84%	116M	0s
## 113050K	84%	102M	0s
## 113100K	84%	103M	0s
## 113150K	84%	85.7M	0s
## 113200K	84%	125M	0s
## 113250K	84%	106M	0s
## 113300K	84%	106M	0s
## 113350K	84%	101M	0s
## 113400K	84%	133M	0s
## 113450K	84%	80.8M	0s
## 113500K	84%	104M	0s
## 113550K	84%	99.5M	0s
## 113600K	84%	108M	0s
## 113650K	84%	110M	0s
## 113700K	84%	118M	0s
## 113750K	84%	103M	0s
## 113800K	84%	109M	0s
## 113850K	84%	122M	0s
## 113900K	84%	106M	0s
## 113950K	85%	99.1M	0s
## 114000K	85%	107M	0s
## 114050K	85%	121M	0s
## 114100K	85%	116M	0s
## 114150K	85%	100M	0s
## 114200K	85%	98.9M	0s
## 114250K	85%	112M	0s
## 114300K	85%	115M	0s
## 114350K	85%	104M	0s
## 114400K	85%	113M	0s
## 114450K	85%	113M	0s
## 114500K	85%	116M	0s

## 114550K	85%	111M	0s
## 114600K	85%	109M	0s
## 114650K	85%	103M	0s
## 114700K	85%	114M	0s
## 114750K	85%	118M	0s
## 114800K	85%	102M	0s
## 114850K	85%	104M	0s
## 114900K	85%	113M	0s
## 114950K	85%	98.0M	0s
## 115000K	85%	111M	0s
## 115050K	85%	117M	0s
## 115100K	85%	122M	0s
## 115150K	85%	93.2M	0s
## 115200K	85%	114M	0s
## 115250K	86%	120M	0s
## 115300K	86%	97.5M	0s
## 115350K	86%	104M	0s
## 115400K	86%	108M	0s
## 115450K	86%	122M	0s
## 115500K	86%	125M	0s
## 115550K	86%	94.4M	0s
## 115600K	86%	121M	0s
## 115650K	86%	119M	0s
## 115700K	86%	110M	0s
## 115750K	86%	112M	0s
## 115800K	86%	104M	0s
## 115850K	86%	132M	0s
## 115900K	86%	117M	0s
## 115950K	86%	93.3M	0s
## 116000K	86%	113M	0s
## 116050K	86%	127M	0s
## 116100K	86%	124M	0s
## 116150K	86%	109M	0s
## 116200K	86%	114M	0s
## 116250K	86%	126M	0s
## 116300K	86%	107M	0s
## 116350K	86%	96.5M	0s
## 116400K	86%	118M	0s
## 116450K	86%	117M	0s
## 116500K	86%	116M	0s
## 116550K	86%	119M	0s
## 116600K	87%	114M	0s
## 116650K	87%	110M	0s
## 116700K	87%	109M	0s
## 116750K	87%	112M	0s
## 116800K	87%	111M	0s
## 116850K	87%	106M	0s
## 116900K	87%	140M	0s
## 116950K	87%	117M	0s
## 117000K	87%	111M	0s
## 117050K	87%	31.7M	0s
## 117100K	87%	117M	0s
## 117150K	87%	120M	0s
## 117200K	87%	40.7M	0s

## 117250K	87%	70.3M	0s
## 117300K	87%	63.8M	0s
## 117350K	87%	95.8M	0s
## 117400K	87%	125M	0s
## 117450K	87%	57.3M	0s
## 117500K	87%	109M	0s
## 117550K	87%	93.8M	0s
## 117600K	87%	112M	0s
## 117650K	87%	107M	0s
## 117700K	87%	111M	0s
## 117750K	87%	110M	0s
## 117800K	87%	111M	0s
## 117850K	87%	144M	0s
## 117900K	87%	102M	0s
## 117950K	88%	85.8M	0s
## 118000K	88%	122M	0s
## 118050K	88%	138M	0s
## 118100K	88%	118M	0s
## 118150K	88%	101M	0s
## 118200K	88%	127M	0s
## 118250K	88%	108M	0s
## 118300K	88%	98.2M	0s
## 118350K	88%	95.9M	0s
## 118400K	88%	116M	0s
## 118450K	88%	117M	0s
## 118500K	88%	128M	0s
## 118550K	88%	105M	0s
## 118600K	88%	121M	0s
## 118650K	88%	102M	0s
## 118700K	88%	122M	0s
## 118750K	88%	101M	0s
## 118800K	88%	116M	0s
## 118850K	88%	120M	0s
## 118900K	88%	111M	0s
## 118950K	88%	110M	0s
## 119000K	88%	109M	0s
## 119050K	88%	106M	0s
## 119100K	88%	113M	0s
## 119150K	88%	97.5M	0s
## 119200K	88%	120M	0s
## 119250K	88%	114M	0s
## 119300K	89%	119M	0s
## 119350K	89%	102M	0s
## 119400K	89%	106M	0s
## 119450K	89%	132M	0s
## 119500K	89%	118M	0s
## 119550K	89%	95.6M	0s
## 119600K	89%	126M	0s
## 119650K	89%	111M	0s
## 119700K	89%	110M	0s
## 119750K	89%	111M	0s
## 119800K	89%	111M	0s
## 119850K	89%	127M	0s
## 119900K	89%	119M	0s

## 119950K	89%	93.1M	0s
## 120000K	89%	97.5M	0s
## 120050K	89%	128M	0s
## 120100K	89%	132M	0s
## 120150K	89%	98.4M	0s
## 120200K	89%	113M	0s
## 120250K	89%	123M	0s
## 120300K	89%	111M	0s
## 120350K	89%	96.0M	0s
## 120400K	89%	110M	0s
## 120450K	89%	104M	0s
## 120500K	89%	115M	0s
## 120550K	89%	127M	0s
## 120600K	89%	106M	0s
## 120650K	90%	117M	0s
## 120700K	90%	128M	0s
## 120750K	90%	97.4M	0s
## 120800K	90%	111M	0s
## 120850K	90%	113M	0s
## 120900K	90%	129M	0s
## 120950K	90%	96.1M	0s
## 121000K	90%	114M	0s
## 121050K	90%	113M	0s
## 121100K	90%	115M	0s
## 121150K	90%	92.2M	0s
## 121200K	90%	135M	0s
## 121250K	90%	118M	0s
## 121300K	90%	103M	0s
## 121350K	90%	90.3M	0s
## 121400K	90%	74.5M	0s
## 121450K	90%	107M	0s
## 121500K	90%	115M	0s
## 121550K	90%	102M	0s
## 121600K	90%	104M	0s
## 121650K	90%	126M	0s
## 121700K	90%	113M	0s
## 121750K	90%	93.9M	0s
## 121800K	90%	124M	0s
## 121850K	90%	126M	0s
## 121900K	90%	122M	0s
## 121950K	91%	102M	0s
## 122000K	91%	114M	0s
## 122050K	91%	101M	0s
## 122100K	91%	108M	0s
## 122150K	91%	108M	0s
## 122200K	91%	110M	0s
## 122250K	91%	113M	0s
## 122300K	91%	139M	0s
## 122350K	91%	109M	0s
## 122400K	91%	107M	0s
## 122450K	91%	116M	0s
## 122500K	91%	104M	0s
## 122550K	91%	109M	0s
## 122600K	91%	129M	0s

## 122650K	91%	112M	0s
## 122700K	91%	95.8M	0s
## 122750K	91%	95.9M	0s
## 122800K	91%	108M	0s
## 122850K	91%	102M	0s
## 122900K	91%	116M	0s
## 122950K	91%	120M	0s
## 123000K	91%	129M	0s
## 123050K	91%	105M	0s
## 123100K	91%	107M	0s
## 123150K	91%	95.6M	0s
## 123200K	91%	124M	0s
## 123250K	91%	123M	0s
## 123300K	92%	129M	0s
## 123350K	92%	98.1M	0s
## 123400K	92%	108M	0s
## 123450K	92%	104M	0s
## 123500K	92%	122M	0s
## 123550K	92%	95.4M	0s
## 123600K	92%	119M	0s
## 123650K	92%	139M	0s
## 123700K	92%	115M	0s
## 123750K	92%	101M	0s
## 123800K	92%	110M	0s
## 123850K	92%	113M	0s
## 123900K	92%	118M	0s
## 123950K	92%	110M	0s
## 124000K	92%	118M	0s
## 124050K	92%	104M	0s
## 124100K	92%	102M	0s
## 124150K	92%	110M	0s
## 124200K	92%	125M	0s
## 124250K	92%	118M	0s
## 124300K	92%	139M	0s
## 124350K	92%	95.6M	0s
## 124400K	92%	115M	0s
## 124450K	92%	104M	0s
## 124500K	92%	107M	0s
## 124550K	92%	109M	0s
## 124600K	92%	140M	0s
## 124650K	93%	104M	0s
## 124700K	93%	122M	0s
## 124750K	93%	86.8M	0s
## 124800K	93%	107M	0s
## 124850K	93%	106M	0s
## 124900K	93%	106M	0s
## 124950K	93%	118M	0s
## 125000K	93%	119M	0s
## 125050K	93%	109M	0s
## 125100K	93%	103M	0s
## 125150K	93%	85.7M	0s
## 125200K	93%	114M	0s
## 125250K	93%	132M	0s
## 125300K	93%	121M	0s

## 125350K	93%	102M	0s
## 125400K	93%	114M	0s
## 125450K	93%	99.5M	0s
## 125500K	93%	108M	0s
## 125550K	93%	88.9M	0s
## 125600K	93%	118M	0s
## 125650K	93%	137M	0s
## 125700K	93%	108M	0s
## 125750K	93%	102M	0s
## 125800K	93%	103M	0s
## 125850K	93%	112M	0s
## 125900K	93%	111M	0s
## 125950K	93%	106M	0s
## 126000K	94%	117M	0s
## 126050K	94%	118M	0s
## 126100K	94%	112M	0s
## 126150K	94%	81.3M	0s
## 126200K	94%	90.3M	0s
## 126250K	94%	82.3M	0s
## 126300K	94%	134M	0s
## 126350K	94%	92.7M	0s
## 126400K	94%	104M	0s
## 126450K	94%	114M	0s
## 126500K	94%	127M	0s
## 126550K	94%	95.0M	0s
## 126600K	94%	129M	0s
## 126650K	94%	114M	0s
## 126700K	94%	107M	0s
## 126750K	94%	100M	0s
## 126800K	94%	106M	0s
## 126850K	94%	97.3M	0s
## 126900K	94%	109M	0s
## 126950K	94%	113M	0s
## 127000K	94%	116M	0s
## 127050K	94%	117M	0s
## 127100K	94%	114M	0s
## 127150K	94%	91.5M	0s
## 127200K	94%	98.8M	0s
## 127250K	94%	110M	0s
## 127300K	94%	95.2M	0s
## 127350K	95%	93.7M	0s
## 127400K	95%	137M	0s
## 127450K	95%	115M	0s
## 127500K	95%	116M	0s
## 127550K	95%	88.7M	0s
## 127600K	95%	108M	0s
## 127650K	95%	104M	0s
## 127700K	95%	123M	0s
## 127750K	95%	102M	0s
## 127800K	95%	114M	0s
## 127850K	95%	126M	0s
## 127900K	95%	112M	0s
## 127950K	95%	84.7M	0s
## 128000K	95%	115M	0s

## 128050K	95%	115M	0s
## 128100K	95%	115M	0s
## 128150K	95%	98.5M	0s
## 128200K	95%	112M	0s
## 128250K	95%	104M	0s
## 128300K	95%	128M	0s
## 128350K	95%	101M	0s
## 128400K	95%	107M	0s
## 128450K	95%	117M	0s
## 128500K	95%	105M	0s
## 128550K	95%	93.4M	0s
## 128600K	95%	109M	0s
## 128650K	95%	107M	0s
## 128700K	96%	121M	0s
## 128750K	96%	79.8M	0s
## 128800K	96%	121M	0s
## 128850K	96%	115M	0s
## 128900K	96%	135M	0s
## 128950K	96%	101M	0s
## 129000K	96%	125M	0s
## 129050K	96%	125M	0s
## 129100K	96%	108M	0s
## 129150K	96%	112M	0s
## 129200K	96%	106M	0s
## 129250K	96%	119M	0s
## 129300K	96%	99.0M	0s
## 129350K	96%	99.5M	0s
## 129400K	96%	127M	0s
## 129450K	96%	140M	0s
## 129500K	96%	103M	0s
## 129550K	96%	86.1M	0s
## 129600K	96%	117M	0s
## 129650K	96%	104M	0s
## 129700K	96%	126M	0s
## 129750K	96%	108M	0s
## 129800K	96%	121M	0s
## 129850K	96%	102M	0s
## 129900K	96%	108M	0s
## 129950K	96%	87.2M	0s
## 130000K	97%	118M	0s
## 130050K	97%	127M	0s
## 130100K	97%	141M	0s
## 130150K	97%	109M	0s
## 130200K	97%	102M	0s
## 130250K	97%	114M	0s
## 130300K	97%	111M	0s
## 130350K	97%	102M	0s
## 130400K	97%	126M	0s
## 130450K	97%	119M	0s
## 130500K	97%	124M	0s
## 130550K	97%	97.6M	0s
## 130600K	97%	114M	0s
## 130650K	97%	118M	0s
## 130700K	97%	110M	0s

## 130750K	97%	116M	0s
## 130800K	97%	112M	0s
## 130850K	97%	113M	0s
## 130900K	97%	107M	0s
## 130950K	97%	109M	0s
## 131000K	97%	113M	0s
## 131050K	97%	115M	0s
## 131100K	97%	124M	0s
## 131150K	97%	104M	0s
## 131200K	97%	101M	0s
## 131250K	97%	104M	0s
## 131300K	97%	92.5M	0s
## 131350K	98%	110M	0s
## 131400K	98%	137M	0s
## 131450K	98%	121M	0s
## 131500K	98%	119M	0s
## 131550K	98%	92.0M	0s
## 131600K	98%	103M	0s
## 131650K	98%	103M	0s
## 131700K	98%	118M	0s
## 131750K	98%	112M	0s
## 131800K	98%	113M	0s
## 131850K	98%	115M	0s
## 131900K	98%	103M	0s
## 131950K	98%	94.2M	0s
## 132000K	98%	114M	0s
## 132050K	98%	117M	0s
## 132100K	98%	125M	0s
## 132150K	98%	98.8M	0s
## 132200K	98%	104M	0s
## 132250K	98%	117M	0s
## 132300K	98%	111M	0s
## 132350K	98%	99.9M	0s
## 132400K	98%	128M	0s
## 132450K	98%	114M	0s
## 132500K	98%	128M	0s
## 132550K	98%	83.5M	0s
## 132600K	98%	95.7M	0s
## 132650K	98%	92.7M	0s
## 132700K	99%	111M	0s
## 132750K	99%	116M	0s
## 132800K	99%	108M	0s
## 132850K	99%	110M	0s
## 132900K	99%	100M	0s
## 132950K	99%	92.1M	0s
## 133000K	99%	127M	0s
## 133050K	99%	139M	0s
## 133100K	99%	142M	0s
## 133150K	99%	116M	0s
## 133200K	99%	147M	0s
## 133250K	99%	143M	0s
## 133300K	99%	139M	0s
## 133350K	99%	129M	0s
## 133400K	99%	135M	0s

```
## 133450K ..... 99% 145M 0s
## 133500K ..... 99% 137M 0s
## 133550K ..... 99% 115M 0s
## 133600K ..... 99% 142M 0s
## 133650K ..... 99% 143M 0s
## 133700K ..... 99% 140M 0s
## 133750K ..... 99% 127M 0s
## 133800K ..... 99% 144M 0s
## 133850K ..... 99% 145M 0s
## 133900K ..... 99% 116M 0s
## 133950K ..... 99% 121M 0s
## 134000K ..... 99% 146M 0s
## 134050K ..... 100% 117M=1.6s
##
## 2021-12-25 15:25:01 (83.6 MB/s) - 'silva_nr99_v138.1_train_set.fa.gz.6' saved [137283333/137283333]
```

```
fastaRef <- "/home/rstudio/silva_nr99_v138.1_train_set.fa.gz"
taxTab <- assignTaxonomy(seqtabNoC, refFasta=fastaRef, multithread=TRUE)
unnname(head(taxTab))
```

```
##      [,1]      [,2]      [,3]      [,4]      [,5]
## [1,] "Bacteria" "Bacteroidota" "Bacteroidia" "Bacteroidales" "Muribaculaceae"
## [2,] "Bacteria" "Bacteroidota" "Bacteroidia" "Bacteroidales" "Muribaculaceae"
## [3,] "Bacteria" "Bacteroidota" "Bacteroidia" "Bacteroidales" "Muribaculaceae"
## [4,] "Bacteria" "Bacteroidota" "Bacteroidia" "Bacteroidales" "Muribaculaceae"
## [5,] "Bacteria" "Bacteroidota" "Bacteroidia" "Bacteroidales" "Bacteroidaceae"
## [6,] "Bacteria" "Bacteroidota" "Bacteroidia" "Bacteroidales" "Muribaculaceae"
##      [,6]
## [1,] NA
## [2,] NA
## [3,] NA
## [4,] NA
## [5,] "Bacteroides"
## [6,] NA
```

#Construire un arbre phylogénétique

```
seqs <- getSequences(seqtabNoC)
names(seqs) <- seqs # This propagates to the tip labels of the tree
alignment <- AlignSeqs(DNAStringSet(seqs), anchor=NA, verbose=FALSE)
```

```
phangAlign <- phyDat(as(alignment, "matrix"), type="DNA")
dm <- dist.ml(phangAlign)
treeNJ <- NJ(dm)
fit = pml(treeNJ, data=phangAlign)
fitGTR <- update(fit, k=4, inv=0.2)
fitGTR <- optim.pml(fitGTR, model="GTR", optInv=TRUE, optGamma=TRUE,
  rearrangement = "stochastic", control = pml.control(trace = 0))
detach("package:phangorn", unload=TRUE)
```

#Combiner les données dnas un objet phyloseq

```
samdf <- read.csv("https://raw.githubusercontent.com/spholmes/F1000_workflow/master/data/MIMARKS_Data_c
samdf$SampleID <- paste0(gsub("00", "", samdf$host_subject_id), "D", samdf$age-21)
samdf <- samdf[!duplicated(samdf$SampleID),]
rownames(seqtabAll) <- gsub("124", "125", rownames(seqtabAll))
all(rownames(seqtabAll) %in% samdf$SampleID)
```

```
## [1] TRUE
```

```
rownames(samdf) <- samdf$SampleID
keep.cols <- c("collection_date", "biome", "target_gene", "target_subfragment",
"host_common_name", "host_subject_id", "age", "sex", "body_product", "tot_mass",
"diet", "family_relationship", "genotype", "SampleID")
samdf <- samdf[rownames(seqtabAll), keep.cols]
```

```
ps <- phyloseq(otu_table(seqtabNoC, taxa_are_rows=FALSE),
sample_data(samdf),
tax_table(taxTab), phy_tree(fitGTR$tree))
ps <- prune_samples(sample_names(ps) != "Mock", ps) # Remove mock sample
ps
```

```
## phyloseq-class experiment-level object
## otu_table() OTU Table: [ 218 taxa and 19 samples ]
## sample_data() Sample Data: [ 19 samples by 14 sample variables ]
## tax_table() Taxonomy Table: [ 218 taxa by 6 taxonomic ranks ]
## phy_tree() Phylogenetic Tree: [ 218 tips and 216 internal nodes ]
```

##Utilisation de phyloseq

##Chargement des données Dans un premier temps, il faut charger les données.

```
ps_connect <-url("https://raw.githubusercontent.com/spholmes/F1000_workflow/master/data/ps.rds")
ps = readRDS(ps_connect)
ps
```

```
## phyloseq-class experiment-level object
## otu_table() OTU Table: [ 389 taxa and 360 samples ]
## sample_data() Sample Data: [ 360 samples by 14 sample variables ]
## tax_table() Taxonomy Table: [ 389 taxa by 6 taxonomic ranks ]
## phy_tree() Phylogenetic Tree: [ 389 tips and 387 internal nodes ]
```

Après avoir importer les données, elles seront filtrées taxonomiquement.

##Filtrage taxonomique

```
rank_names(ps)
```

```
## [1] "Kingdom" "Phylum" "Class" "Order" "Family" "Genus"
```

```
table(tax_table(ps)[, "Phylum"], exclude = NULL)
```

```
##
##           Actinobacteria           Bacteroidetes
##                13                23
## Candidatus_Saccharibacteria   Cyanobacteria/Chloroplast
##                1                4
##           Deinococcus-Thermus           Firmicutes
##                1                327
##           Fusobacteria           Proteobacteria
##                1                11
##           Tenericutes           Verrucomicrobia
##                1                1
##                <NA>
##                6
```

```
ps <- subset_taxa(ps, !is.na(Phylum) & !Phylum %in% c("", "uncharacterized"))
```

```
prevdf = apply(X = otu_table(ps),
               MARGIN = ifelse(taxa_are_rows(ps), yes = 1, no = 2),
               FUN = function(x){sum(x > 0)})
prevdf = data.frame(Prevalence = prevdf,
                   TotalAbundance = taxa_sums(ps),
                   tax_table(ps))
```

```
plyr::ddply(prevdf, "Phylum", function(df1){cbind(mean(df1$Prevalence),sum(df1$Prevalence))})
```

```
##           Phylum           1           2
## 1 Actinobacteria 120.15385 1562
## 2 Bacteroidetes 265.52174 6107
## 3 Candidatus_Saccharibacteria 280.00000 280
## 4 Cyanobacteria/Chloroplast 64.25000 257
## 5 Deinococcus-Thermus 52.00000 52
## 6 Firmicutes 179.24771 58614
## 7 Fusobacteria 2.00000 2
## 8 Proteobacteria 59.09091 650
## 9 Tenericutes 234.00000 234
## 10 Verrucomicrobia 104.00000 104
```

```
filterPhyla = c("Fusobacteria", "Deinococcus-Thermus")
ps1 = subset_taxa(ps, !Phylum %in% filterPhyla)
ps1
```

```
## phyloseq-class experiment-level object
## otu_table() OTU Table: [ 381 taxa and 360 samples ]
## sample_data() Sample Data: [ 360 samples by 14 sample variables ]
## tax_table() Taxonomy Table: [ 381 taxa by 6 taxonomic ranks ]
## phy_tree() Phylogenetic Tree: [ 381 tips and 379 internal nodes ]
```

##Filtration de la prévalence

```
prevdf1 = subset(prevdf, Phylum %in% get_taxa_unique(ps1, "Phylum"))
ggplot(prevdf1, aes(TotalAbundance, Prevalence / nsamples(ps),color=Phylum)) +
```



```
geom_hline(yintercept = 0.05, alpha = 0.5, linetype = 2) + geom_point(size = 2, alpha = 0.7) +
scale_x_log10() + xlab("Total Abundance") + ylab("Prevalence [Frac. Samples]") +
facet_wrap(~Phylum) + theme(legend.position="none")
```

ecog-analyses_files/figure-latex/unnamed-chunk-38-1.pdf

```
prevalenceThreshold = 0.05 * nsamples(ps)
prevalenceThreshold
```

```
## [1] 18
```

```
keepTaxa = rownames(prevdf1)[(prevdf1$Prevalence >= prevalenceThreshold)]
ps2 = prune_taxa(keepTaxa, ps)
```

```
##Taxons agglomérés
```

```
length(get_taxa_unique(ps2, taxonomic.rank = "Genus"))
```

```
## [1] 49
```

```
ps3 = tax_glom(ps2, "Genus", NArm = TRUE)
```

```
h1 = 0.4
ps4 = tip_glom(ps2, h = h1)
```

```
multiPlotTitleTextSize = 15
p2tree = plot_tree(ps2, method = "treeonly",
                    ladderize = "left",
                    title = "Before Agglomeration") +
  theme(plot.title = element_text(size = multiPlotTitleTextSize))
p3tree = plot_tree(ps3, method = "treeonly",
                    ladderize = "left", title = "By Genus") +
  theme(plot.title = element_text(size = multiPlotTitleTextSize))
p4tree = plot_tree(ps4, method = "treeonly",
                    ladderize = "left", title = "By Height") +
  theme(plot.title = element_text(size = multiPlotTitleTextSize))
```

```
grid.arrange(nrow = 1, p2tree, p3tree, p4tree)
```

ecog-analyses_files/figure-latex/unnamed-chunk-45-1.pdf

```
##Transformation de la valeur d'abondance
```

```
plot_abundance = function(physeq,title = "",
                          Facet = "Order", Color = "Phylum"){
  p1f = subset_taxa(physeq, Phylum %in% c("Firmicutes"))
  mphyseq = psmelt(p1f)
  mphyseq <- subset(mphyseq, Abundance > 0)
  ggplot(data = mphyseq, mapping = aes_string(x = "sex",y = "Abundance",
                                              color = Color, fill = Color)) +
    geom_violin(fill = NA) +
    geom_point(size = 1, alpha = 0.3,
              position = position_jitter(width = 0.3)) +
    facet_wrap(facets = Facet) + scale_y_log10()+
    theme(legend.position="none")
}
```

```
ps3ra = transform_sample_counts(ps3, function(x){x / sum(x)})
```

```
plotBefore = plot_abundance(ps3,"")
plotAfter = plot_abundance(ps3ra,"")

grid.arrange(nrow = 2, plotBefore, plotAfter)
```

ecog-analyses_files/figure-latex/unnamed-chunk-48-1.pdf

##Sous-ensemble par taxonomie

```
ps0rd = subset_taxa(ps3ra, Order == "Lactobacillales")
plot_abundance(ps0rd, Facet = "Genus", Color = NULL)
```

ecog-analyses_files/figure-latex/unnamed-chunk-49-1.pdf

Installation de packages necessaires pour les analyses complémentaires.

```
.cran_packages <- c( "shiny","miniUI", "caret", "pls", "e1071", "ggplot2", "randomForest", "dplyr", "ggplot2",
                    "reshape2", "PMA", "structSSI", "ade4",
                    "ggnetwork", "intergraph", "scales")
.github_packages <- c("jfkuyama/phyloseqGraphTest")
.bioc_packages <- c("genefilter", "impute")
# Install CRAN packages (if not already installed)
.inst <- .cran_packages %in% installed.packages()
if (any(!.inst)){
  install.packages(.cran_packages[!.inst],repos = "http://cran.rstudio.com/")
}
```

```
## Installing package into '/usr/local/lib/R/site-library'
## (as 'lib' is unspecified)

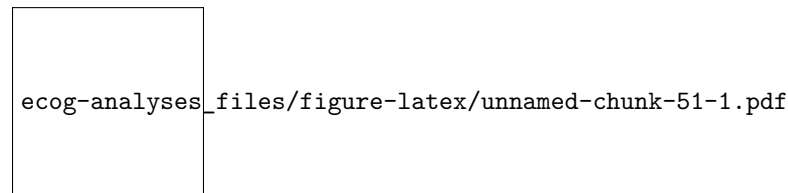
## Warning: package 'structSSI' is not available for this version of R
##
## A version of this package for your version of R might be available elsewhere,
## see the ideas at
## https://cran.r-project.org/doc/manuals/r-patched/R-admin.html#Installing-packages
```

```
.inst <- .github_packages %in% installed.packages()
if (any(!.inst)){
  devtools::install_github(.github_packages[!.inst])
}
```

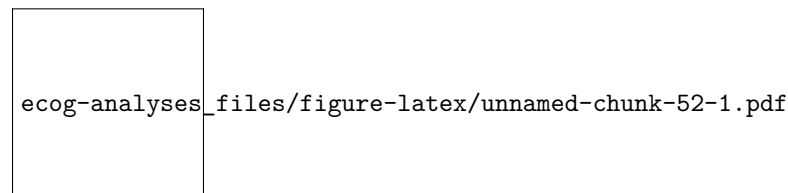
```
## Skipping install of 'phyloseqGraphTest' from a github remote, the SHA1 (3fb6c274) has not changed since
## Use 'force = TRUE' to force installation
```

```
##TPrétraitement
```

```
qplot(sample_data(ps)$age, geom = "histogram", binwidth=20) + xlab("age")
```



```
qplot(log10(rowSums(otu_table(ps))), binwidth=0.2) +
  xlab("Logged counts-per-sample")
```



```
sample_data(ps)$age_binned <- cut(sample_data(ps)$age,
                                   breaks = c(0, 100, 200, 400))
levels(sample_data(ps)$age_binned) <- list(Young100="(0,100]", Mid100to200="(100,200]", Old200="(200,400]")
sample_data(ps)$family_relationship=gsub(" ", "", sample_data(ps)$family_relationship)
pslog <- transform_sample_counts(ps, function(x) log(1 + x))
out.wuf.log <- ordinate(pslog, method = "MDS", distance = "wunifrac")
```

```
## Warning in UniFrac(physeq, weighted = TRUE, ...): Randomly assigning root as --
## GCAAGCGTTATCCGGAATGACTGGGCGTAAAGGGTGCGTAGGTGTTTGGCAAGTTGGTAGCGTAATCCGGGGCTCAACCTCGGCGCTACTACCAAAC
## -- in the phylogenetic tree in the data you provided.
```

```
evals <- out.wuf.log$values$Eigenvalues
plot_ordination(pslog, out.wuf.log, color = "age_binned") +
  labs(col = "Binned Age") +
  coord_fixed(sqrt(evals[2] / evals[1]))
```

ecog-analyses_files/figure-latex/unnamed-chunk-53-1.pdf

```
rel_abund <- t(apply(otu_table(ps), 1, function(x) x / sum(x)))
qplot(rel_abund[, 12], geom = "histogram", binwidth=0.05) +
  xlab("Relative abundance")
```

ecog-analyses_files/figure-latex/unnamed-chunk-54-1.pdf

#Différentes projections d'ordination

```
outliers <- c("F5D165", "F6D165", "M3D175", "M4D175", "M5D175", "M6D175")
ps <- prune_samples(!(sample_names(ps) %in% outliers), ps)
```

```
which(!rowSums(otu_table(ps)) > 1000)
```

```
## F5D145 M1D149 M1D9 M2D125 M2D19 M3D148 M3D149 M3D3 M3D5 M3D8
##      69   185   200   204   218   243   244   252   256   260
```

```
ps <- prune_samples(rowSums(otu_table(ps)) > 1000, ps)
pslog <- transform_sample_counts(ps, function(x) log(1 + x))
```

```
out.pcoa.log <- ordinate(pslog, method = "MDS", distance = "bray")
evals <- out.pcoa.log$values[,1]
plot_ordination(pslog, out.pcoa.log, color = "age_binned",
  shape = "family_relationship") +
  labs(col = "Binned Age", shape = "Litter") +
  coord_fixed(sqrt(evals[2] / evals[1]))
```

ecog-analyses_files/figure-latex/unnamed-chunk-58-1.pdf

```
out.dpcoa.log <- ordinate(pslog, method = "DPCoA")
evals <- out.dpcoa.log$eig
plot_ordination(pslog, out.dpcoa.log, color = "age_binned", label= "SampleID",
                shape = "family_relationship") +
  labs(col = "Binned Age", shape = "Litter")+
  coord_fixed(sqrt(evals[2] / evals[1]))
```

ecog-analyses_files/figure-latex/unnamed-chunk-59-1.pdf

```
plot_ordination(pslog, out.dpcoa.log, type = "species", color = "Phylum") +
  coord_fixed(sqrt(evals[2] / evals[1]))
```

ecog-analyses_files/figure-latex/unnamed-chunk-60-1.pdf

```
out.wuf.log <- ordinate(pslog, method = "PCoA", distance = "wunifrac")
```

```
## Warning in UniFrac(physeq, weighted = TRUE, ...): Randomly assigning root as --
## GCAAGCGTTATCCGGATTACTGGGTGTAAAGGGAGCGTAGACGGCTTGATAAGTCTGAAGTGAAAGGCCAAGGCTTAACCATGGAAGTCTTTGGAAAC
## -- in the phylogenetic tree in the data you provided.
```

```
evals <- out.wuf.log$values$Eigenvalues
plot_ordination(pslog, out.wuf.log, color = "age_binned",
                shape = "family_relationship") +
  coord_fixed(sqrt(evals[2] / evals[1])) +
  labs(col = "Binned Age", shape = "Litter")
```

ecog-analyses_files/figure-latex/unnamed-chunk-61-1.pdf

#Pourquoi les parcelles d'ordination sont-elles si loin d'être carrées ?

##APC sur les rangs

```
abund <- otu_table(pslog)
abund_ranks <- t(apply(abund, 1, rank))
```

```

bund_ranks <- abund_ranks - 329
abund_ranks[abund_ranks < 1] <- 1

library(dplyr)

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:Biostrings':
##
## collapse, intersect, setdiff, setequal, union

## The following object is masked from 'package:GenomeInfoDb':
##
## intersect

## The following object is masked from 'package:XVector':
##
## slice

## The following objects are masked from 'package:IRanges':
##
## collapse, desc, intersect, setdiff, slice, union

## The following objects are masked from 'package:S4Vectors':
##
## first, intersect, rename, setdiff, setequal, union

## The following objects are masked from 'package:BiocGenerics':
##
## combine, intersect, setdiff, union

## The following object is masked from 'package:gridExtra':
##
## combine

## The following objects are masked from 'package:stats':
##
## filter, lag

## The following objects are masked from 'package:base':
##
## intersect, setdiff, setequal, union

library(reshape2)
abund_df <- melt(abund, value.name = "abund") %>%
  left_join(melt(abund_ranks, value.name = "rank"))

## Joining, by = c("Var1", "Var2")

```

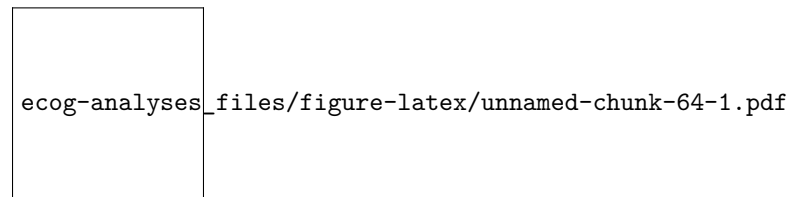
```
colnames(abund_df) <- c("sample", "seq", "abund", "rank")
```

```
abund_df <- melt(abund, value.name = "abund") %>%
  left_join(melt(abund_ranks, value.name = "rank"))
```

```
## Joining, by = c("Var1", "Var2")
```

```
colnames(abund_df) <- c("sample", "seq", "abund", "rank")
```

```
sample_ix <- sample(1:nrow(abund_df), 8)
ggplot(abund_df %>%
  filter(sample %in% abund_df$sample[sample_ix])) +
  geom_point(aes(x = abund, y = rank, col = sample),
    position = position_jitter(width = 0.2), size = 1.5) +
  labs(x = "Abundance", y = "Thresholded rank") +
  scale_color_brewer(palette = "Set2")
```



ecog-analyses_files/figure-latex/unnamed-chunk-64-1.pdf

```
library(ade4)
```

```
##
```

```
## Attaching package: 'ade4'
```

```
## The following object is masked from 'package:Biostrings':
```

```
##
```

```
## score
```

```
## The following object is masked from 'package:BiocGenerics':
```

```
##
```

```
## score
```

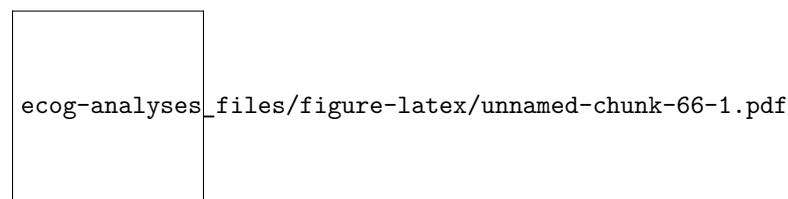
```
ranks_pca <- dudi.pca(abund_ranks, scannf = F, nf = 3)
row_scores <- data.frame(li = ranks_pca$li,
  SampleID = rownames(abund_ranks))
col_scores <- data.frame(co = ranks_pca$co,
  seq = colnames(abund_ranks))
tax <- tax_table(ps) %>%
  data.frame(stringsAsFactors = FALSE)
tax$seq <- rownames(tax)
main_orders <- c("Clostridiales", "Bacteroidales", "Lactobacillales",
  "Coriobacteriales")
tax$Order[!(tax$Order %in% main_orders)] <- "Other"
tax$Order <- factor(tax$Order, levels = c(main_orders, "Other"))
tax$otu_id <- seq_len(ncol(otu_table(ps)))
row_scores <- row_scores %>%
  left_join(sample_data(pslog))
```

```
## Joining, by = "SampleID"
```

```
col_scores <- col_scores %>%  
  left_join(tax)
```

```
## Joining, by = "seq"
```

```
evals_prop <- 100 * (ranks_pca$eig / sum(ranks_pca$eig))  
ggplot() +  
  geom_point(data = row_scores, aes(x = li.Axis1, y = li.Axis2), shape = 2) +  
  geom_point(data = col_scores, aes(x = 25 * co.Comp1, y = 25 * co.Comp2, col = Order),  
            size = .3, alpha = 0.6) +  
  scale_color_brewer(palette = "Set2") +  
  facet_grid(~ age_binned) +  
  guides(col = guide_legend(override.aes = list(size = 3))) +  
  labs(x = sprintf("Axis1 [%s%% variance]", round(evals_prop[1], 2)),  
       y = sprintf("Axis2 [%s%% variance]", round(evals_prop[2], 2))) +  
  coord_fixed(sqrt(ranks_pca$eig[2] / ranks_pca$eig[1])) +  
  theme(panel.border = element_rect(color = "#787878", fill = alpha("white", 0)))
```



```
## Correspondance canonique
```

```
ps_ccpna <- ordinate(pslog, "CCA", formula = pslog ~ age_binned + family_relationship)
```

```
library(ggrepel)  
ps_scores <- vegan::scores(ps_ccpna)  
sites <- data.frame(ps_scores$sites)  
sites$SampleID <- rownames(sites)  
sites <- sites %>%  
  left_join(sample_data(ps))
```

```
## Joining, by = "SampleID"
```

```
species <- data.frame(ps_scores$species)  
species$otu_id <- seq_along(colnames(otu_table(ps)))  
species <- species %>%  
  left_join(tax)
```

```
## Joining, by = "otu_id"
```

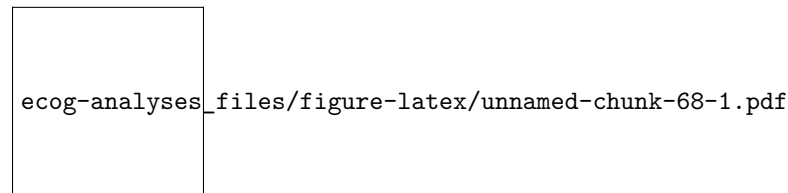
```
evals_prop <- 100 * ps_ccpna$CCA$eig[1:2] / sum(ps_ccpna$CA$eig)  
ggplot() +  
  geom_point(data = sites, aes(x = CCA1, y = CCA2), shape = 2, alpha = 0.5) +  
  geom_point(data = species, aes(x = CCA1, y = CCA2, col = Order), size = 0.5) +
```



```
geom_text_repel(data = species %>% filter(CCA2 < -2),
               aes(x = CCA1, y = CCA2, label = otu_id),
               size = 1.5, segment.size = 0.1) +
facet_grid(. ~ family_relationship) +
guides(col = guide_legend(override.aes = list(size = 3))) +
labs(x = sprintf("Axis1 [%s%% variance]", round(evals_prop[1], 2)),
     y = sprintf("Axis2 [%s%% variance]", round(evals_prop[2], 2))) +
scale_color_brewer(palette = "Set2") +
coord_fixed(sqrt(ps_ccpna$CCA$eig[2] / ps_ccpna$CCA$eig[1])*0.45 ) +
theme(panel.border = element_rect(color = "#787878", fill = alpha("white", 0)))
```

```
## Warning: ggrepel: 9 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```

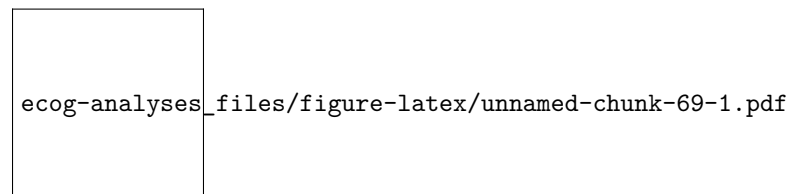
```
## Warning: ggrepel: 9 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```



This is an R Markdown Notebook. When you execute code within the notebook, the results appear beneath the code.

Try executing this chunk by clicking the *Run* button within the chunk or by placing your cursor inside it and pressing *Ctrl+Shift+Enter*.

```
plot(cars)
```



Add a new chunk by clicking the *Insert Chunk* button on the toolbar or by pressing *Ctrl+Alt+I*.

When you save the notebook, an HTML file containing the code and output will be saved alongside it (click the *Preview* button or press *Ctrl+Shift+K* to preview the HTML file).

The preview shows you a rendered HTML copy of the contents of the editor. Consequently, unlike *Knit*, *Preview* does not run any R code chunks. Instead, the output of the chunk when it was last run in the editor is displayed.