PWS470 final

Brian Colgrove

due April 21, 2021, 1159PM

Genome Assembly

S thermophilus genome (6 pts)

Overview paragraph

```
print("The number of reads in the raw sequence files is 2160000. The estimated size of the genom
e is 1.8 megabases.The estimated nucleotide coverage of the genome with the raw sequence files i
s 227. It is necessary to split the genome. The kmers range I used in my assembly was 91 to 131
 by 8. The ideal kmer to use was 107. My velve-estimated exp_cov was 26 and my histogram-estimat
ed cov_cutoff was 23 The manual gave better statistics")
```

```
## [1] "The number of reads in the raw sequence files is 2160000. The estimated size of the geno
me is 1.8 megabases.The estimated nucleotide coverage of the genome with the raw sequence files
is 227. It is necessary to split the genome. The kmers range I used in my assembly was 91 to 131
by 8. The ideal kmer to use was 107. My velve-estimated exp_cov was 26 and my histogram-estimate
d cov_cutoff was 23 The manual gave better statistics"
```

Print info on nucleotide coverage in raw reads

```
# Print the number of lines in one of the sequence files in the ~/quizzes/FINAL/ folder.
#cd quizzes/FINAL
expr $(cat Sthermophilus1058_00for.fastq | wc -l)
```

```
## 8640000
```

```
# State the expected size of your genome
print("My expected genome size is 1.8 megabases")
```

```
## [1] "My expected genome size is 1.8 megabases"
```

Print info for splitting genome, if necessary

```
# If necessary, split the genome. If so, store the split file in your ~/velvet_1.2.10/data folde
r, show the number of lines in the file, and re-esimate nucleotide coverage. If it takes you mul
tiple tries you do not need to document all your error-tries; just show the time it worked. Make
sure to show all code for the time that got you between kmer coverage 20 and 30.
split -a 2 -l 5709252 Sthermophilus1058_00for.fastq --additional-suffix=Sthermo_for.fastq
split -a 2 -l 5709252 Sthermophilus1058_00for.fastq --additional-suffix=Sthermo_rev.fastq
```

```
# print the number of lines in your split file
expr $(cat xaaSthermo_for.fastq | wc -l)
```

```
## 5709252
```

```
# Print the estimated nucleotide coverage from your split file
print("113")
```

```
## [1] "113"
```

run velvet

```
velvet_1.2.10/velveth thermo 91,131,8 -fastq -shortPaired  xaaSthermo_for.fastq xaaSthermo_rev.f
astq

for i in {91..123..8}; do velvet_1.2.10/velvetg thermo_$i -cov_cutoff auto -ins_length 450 -exp_
cov auto -read_trkg yes; done;
```

Print table with assembly statistics

```
tail -n 1 thermo_*/Log | sed ':a;N;$!ba;s/\n//g' | sed 's/thermo/\n/g' | cut -f 1,5,10,12,14 -d
" " --output-delimiter '  ' | sed 's/,//g' | sed 's/Log//g' | sed 's/_//g' | sed 's/==>//g' | tr
-d '/' | sort -k1 -n
```

```
##
## 91  336  47724  153501  1772695
## 99  334  47732  153509  1775066
## 107  244  49414  107119  1316214
## 115
## 123  0  0  0  0
```

Create histogram; estimate cov_cutoff and exp_cov

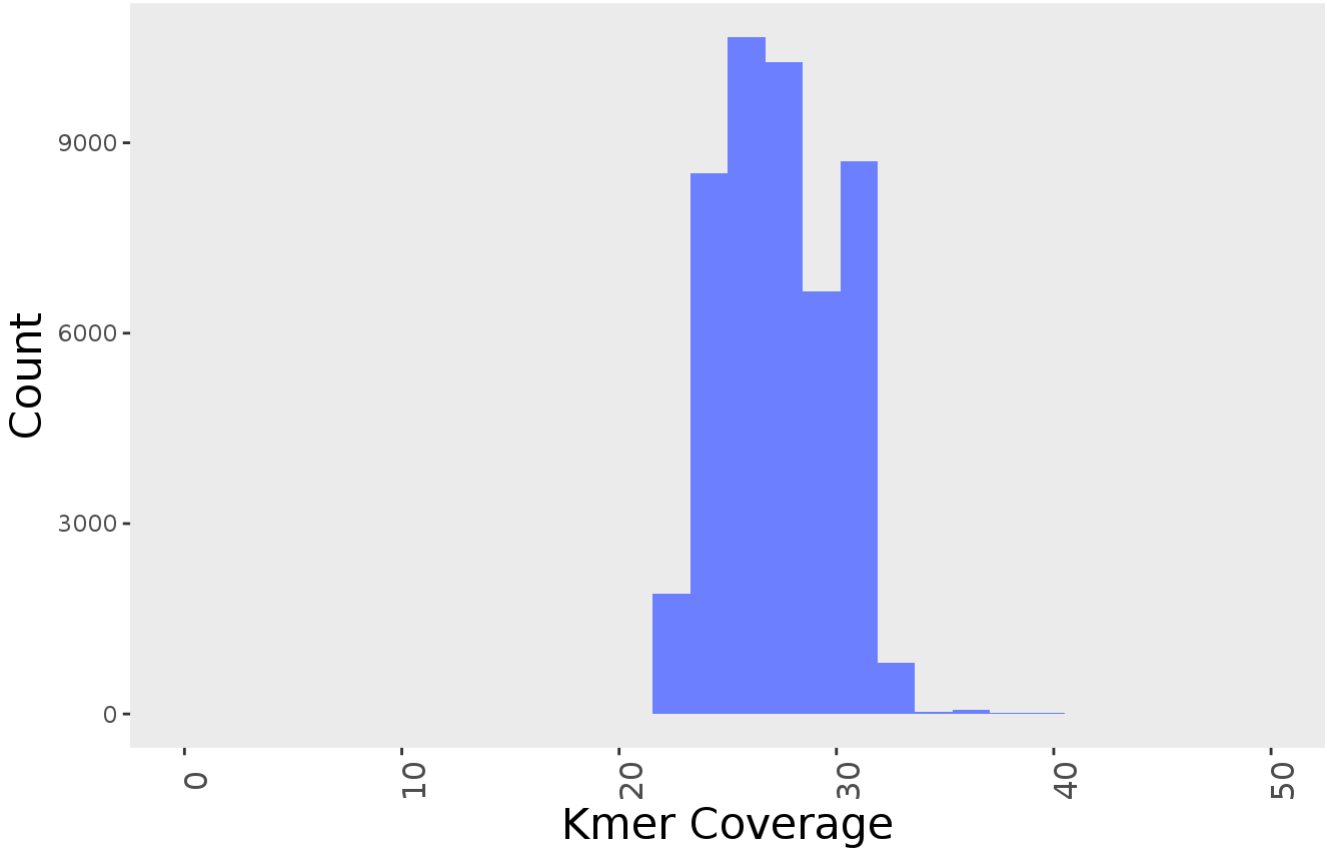
```
data_107 <- read.table('thermo_107/stats.txt', header = T, sep = "\t", fill = T) %>% mutate(cum_
lgth = cumsum(lgth))
ggplot(data_107, aes(short1_cov,weight = lgth/short1_cov)) + geom_histogram(fill = "#6b7fff") +
 xlim(0,50) + labs(title = "Histogram of kmers") +
  theme(title = element_text(size = 24), axis.title.x = element_text(size = 16), axis.title.y =
 element_text(size = 16), axis.text.x = element_text(size = 12, angle = 90), legend.position =
"none", panel.grid = element_blank()) +
  xlab("Kmer Coverage") +
  ylab("Count")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 105 rows containing non-finite values (stat_bin).
```

```
## Warning: Removed 2 rows containing missing values (geom_bar).
```

Histogram of kmers



Rerun velvetg

```
velvet_1.2.10/velvetg thermo_107 -cov_cutoff 23 -ins_length 450 -exp_cov 26 -read_trkg yes
```

Print table comparing auto and manual

```
tail -n 24 thermo_107/Log | sed ':a;N;$!ba;s/\n//g' | sed 's/Final/\n/g' | grep -n 'graph' |cut
 -f 1,9,11,13 -d " " --output-delimiter ' ' | sed 's/,/\t/g'  | sed 's/2: /auto /g' |  sed 's/3:
/manual /g'
```

```
## auto 39270    107117  1778701
## manual 49414  107119  1316214
```

L rhamnosus genome (6 pts)

Overview paragraph

```
print("The number of reads in the raw sequence files is 1800000. The estimated size of the genom
e is 3 megabases.The estimated nucleotide coverage of the genome with the raw sequence files is
 150. It is not necessary to split the genome. The kmers range I used in my assembly was 89 to 1
21 by 8. The ideal kmer to use was 97. My velve-estimated exp_cov was 25 and my histogram-estima
ted cov_cutoff was 20. The manual gave better statistics")
```

```
## [1] "The number of reads in the raw sequence files is 1800000. The estimated size of the geno
me is 3 megabases.The estimated nucleotide coverage of the genome with the raw sequence files is
150. It is not necessary to split the genome. The kmers range I used in my assembly was 89 to 12
1 by 8. The ideal kmer to use was 97. My velve-estimated exp_cov was 25 and my histogram-estimat
ed cov_cutoff was 20. The manual gave better statistics"
```

Print info on nucleotide coverage in raw reads

```
# Print the number of lines in one of the sequence files in the ~/quizzes/FINAL/ folder.
#cd quizzes/FINAL
expr $(cat Lrhamnosus1107_00for.fastq | wc -l)
```

```
## 7200000
```

```
# State the expected size of your genome
print("My expected genome size is 3 megabases")
```

```
## [1] "My expected genome size is 3 megabases"
```

Print info for splitting genome, if necessary

```
# If necessary, split the genome. If so, store the split file in your ~/velvet_1.2.10/data folde
r, show the number of lines in the file, and re-esimate nucleotide coverage. If it takes you mul
tiple tries you do not need to document all your error-tries; just show the time it worked. Make
sure to show all code for the time that got you between kmer coverage 20 and 30.
```

```
# print the number of lines in your split file
```

```
# Print the estimated nucleotide coverage from your split file
print("Does not need to be split")
```

```
## [1] "Does not need to be split"
```

run velvet

```
velvet_1.2.10/velveth lrham 89,121,8 -fastq -shortPaired  Lrhamnosus1107_00for.fastq Lrhamnosus1
107_00rev.fastq

for i in {89..113..8}; do velvet_1.2.10/velvetg lrham_$i -cov_cutoff auto -ins_length 450 -exp_c
ov auto -read_trkg yes; done;
```

Print table with assembly statistics

```
tail -n 1 lrham_*/Log | sed ':a;N;$!ba;s/\n//g' | sed 's/lrham/\n/g' | cut -f 1,5,10,12,14 -d "
 " --output-delimiter '  ' | sed 's/,//g' | sed 's/Log//g' | sed 's/_//g' | sed 's/==>//g' | tr
 -d '/' | sort -k1 -n
```

```
##
## 89  161  95808  434061  2887656
## 97  117  102469  318299  2730709
## 105  191  105536  328518  2889837
## 113  419  20660  121546  2887214
```

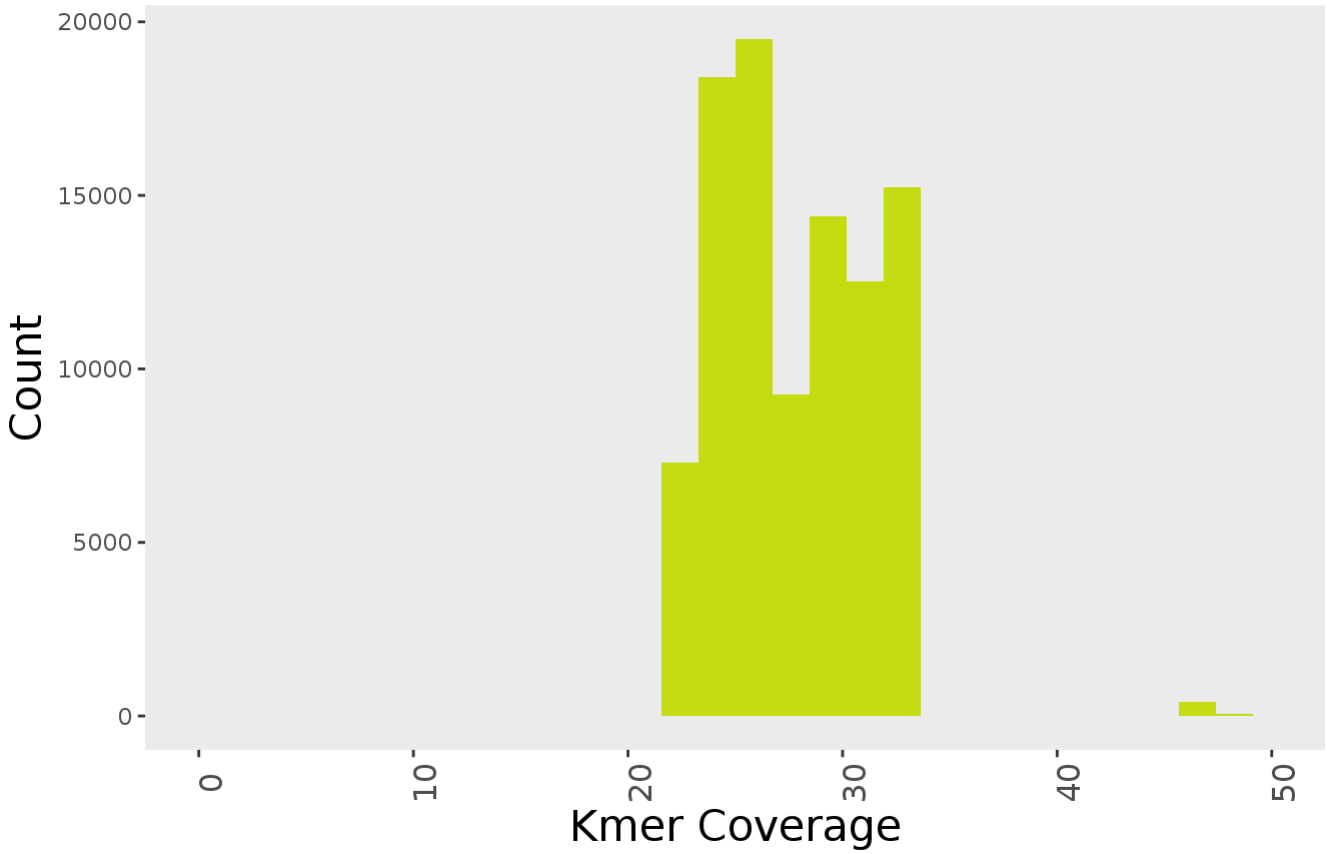
Create histogram; estimate cov_cutoff and exp_cov

```
data_97 <- read.table('lrham_97/stats.txt', header = T, sep = "\t", fill = T) %>% mutate(cum_lgt
h = cumsum(lgth))
ggplot(data_97, aes(short1_cov,weight = lgth/short1_cov)) + geom_histogram(fill = "#c3db0f") + x
lim(0,50) + labs(title = "Histogram of kmers") +
  theme(title = element_text(size = 24), axis.title.x = element_text(size = 16), axis.title.y =
 element_text(size = 16), axis.text.x = element_text(size = 12, angle = 90), legend.position =
"none", panel.grid = element_blank()) +
  xlab("Kmer Coverage") +
  ylab("Count")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 51 rows containing non-finite values (stat_bin).
```

```
## Warning: Removed 2 rows containing missing values (geom_bar).
```



Rerun velvetg

```
velvet_1.2.10/velvetg lrham_97 -cov_cutoff 23 -ins_length 450 -exp_cov 28 -read_trkg yes
```

Print table comparing auto and manual

```
#tail -n 24 lrham_97/Log | cut -f 3,13,15,24 -d $'\n' | cut -f 4,5,9,11,13 -d ' ' | sed 'N;s/\n/
/' | cut -f 2,5,7,8,9 -d ' ' | sed 's/\,//g' | sed 's/23 /manual /g' | sed 's/ /\t/g'

tail -n 24 lrham_97/Log | sed ':a;N;$!ba;s/\n//g' | sed 's/Final/\n/g' | grep -n 'graph' |cut -f
1,9,11,13 -d " " --output-delimiter ' ' | sed 's/,/\t/g'  | sed 's/2: /auto /g' |  sed 's/3: /ma
nual /g'
```

```
## auto 96824    434069  2888164
## manual 102469     318299  2730709
```

L plantarum genome (6 pts)

Overview paragraph

```
print("The number of reads in the raw sequence files is 1800000. The estimated size of the genom
e is 3.25 megabases.The estimated nucleotide coverage of the genome with the raw sequence files
 is 138. It is not necessary to split the genome. The kmers range I used in my assembly was 89 t
o 123 by 8. The ideal kmer to use was 97. My velve-estimated exp_cov was 25 and my histogram-est
imated cov_cutoff was 21. The auto gave better statistics")
```

```
## [1] "The number of reads in the raw sequence files is 1800000. The estimated size of the geno
me is 3.25 megabases.The estimated nucleotide coverage of the genome with the raw sequence files
is 138. It is not necessary to split the genome. The kmers range I used in my assembly was 89 to
123 by 8. The ideal kmer to use was 97. My velve-estimated exp_cov was 25 and my histogram-estim
ated cov_cutoff was 21. The auto gave better statistics"
```

Print info on nucleotide coverage in raw reads

```
# Print the number of lines in one of the sequence files in the ~/quizzes/FINAL/ folder.
#cd quizzes/FINAL
expr $(cat Lplantarum1108_00for.fastq | wc -l)
```

```
## 7200000
```

```
# State the expected size of your genome
print("My expected genome size is 3.25 megabases")
```

```
## [1] "My expected genome size is 3.25 megabases"
```

Print info for splitting genome, if necessary

```
# If necessary, split the genome. If so, store the split file in your ~/velvet_1.2.10/data folde
r, show the number of lines in the file, and re-esimate nucleotide coverage. If it takes you mul
tiple tries you do not need to document all your error-tries; just show the time it worked. Make
sure to show all code for the time that got you between kmer coverage 20 and 30.
```

```
# print the number of lines in your split file
```

```
# Print the estimated nucleotide coverage from your split file
print("Does not need to be split")
```

```
## [1] "Does not need to be split"
```

run velvet

```
velvet_1.2.10/velveth lplant 89,123,8 -fastq -shortPaired  Lplantarum1108_00for.fastq Lplantarum
1108_00rev.fastq

for i in {89..121..8}; do velvet_1.2.10/velvetg lplant_$i -cov_cutoff auto -ins_length 450 -exp_
cov auto -read_trkg yes; done;
```

Print table with assembly statistics

```
tail -n 1 lplant_*/Log | sed ':a;N;$!ba;s/\n//g' | sed 's/lplant/\n/g' | cut -f 1,5,10,12,14 -d
" " --output-delimiter '  ' | sed 's/,//g' | sed 's/Log//g' | sed 's/_//g' | sed 's/==>//g' | tr
-d '/' | sort -k1 -n
```

```
##
## 89
## 97  206  82771  223097  2498123
## 105  284  80228  224484  3330519
## 113  881  10152  186135  3309947
## 121
```

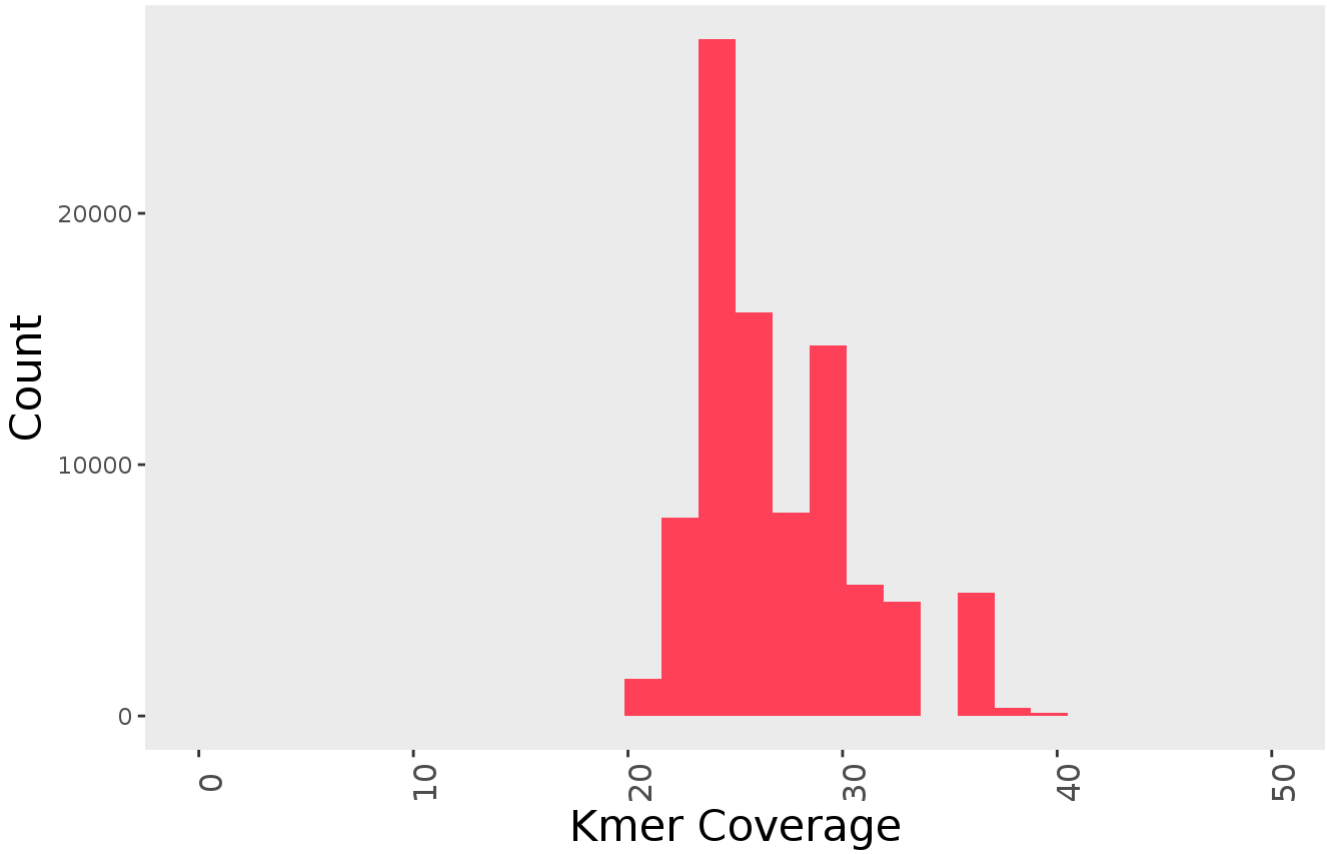
Create histogram; estimate cov_cutoff and exp_cov

```
data_97 <- read.table('lplant_97/stats.txt', header = T, sep = "\t", fill = T) %>% mutate(cum_lg
th = cumsum(lgth))
ggplot(data_97, aes(short1_cov,weight = lgth/short1_cov )) + geom_histogram(fill = "#ff4059") +
 xlim(0,50) +
  labs(title = "Histogram of kmers") +
  theme(title = element_text(size = 24), axis.title.x = element_text(size = 16), axis.title.y =
 element_text(size = 16), axis.text.x = element_text(size = 12, angle = 90), legend.position =
"none", panel.grid = element_blank()) +
  xlab("Kmer Coverage") +
  ylab("Count")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 93 rows containing non-finite values (stat_bin).
```

```
## Warning: Removed 2 rows containing missing values (geom_bar).
```



Rerun velvetg

```
velvet_1.2.10/velvetg lplant_97 -cov_cutoff 21 -ins_length 450 -exp_cov 25 -read_trkg yes
```

Print table comparing auto and manual

```
#tail -n 24 lplant_97/Log | cut -f 3,13,15,24 -d $'\n' | cut -f 4,5,9,11,13 -d ' ' | sed 'N;s/
\n/ /' | cut -f 2,5,7,8,9 -d ' ' | sed 's/\,//g' | sed 's/21 /manual /g' | sed 's/ /\t/g'

tail -n 24 lplant_97/Log | sed ':a;N;$!ba;s/\n//g' | sed 's/Final/\n/g' | grep -n 'graph' |cut -
f 1,9,11,13 -d " " --output-delimiter ' ' | sed 's/,/\t/g'  | sed 's/2: /auto /g' |  sed 's/3: /
manual /g'
```

```
## auto 91580    223097  3328754
## manual 82771  223097  2498123
```

L mesenteroides genome (6 pts)

Overview paragraph

```
print("The number of reads in the raw sequence files is 3326400 The estimated size of the genome
is 3.25 megabases.The estimated nucleotide coverage of the genome with the raw sequence files is
412. It is necessary to split the genome. The kmers range I used in my assembly was 51 to 83 by
 8. The ideal kmer to use was 67. My velve-estimated exp_cov was 21 and my histogram-estimated c
ov_cutoff was 16 The auto gave better statistics")
```

```
## [1] "The number of reads in the raw sequence files is 3326400 The estimated size of the genom
e is 3.25 megabases.The estimated nucleotide coverage of the genome with the raw sequence files
is 412. It is necessary to split the genome. The kmers range I used in my assembly was 51 to 83
by 8. The ideal kmer to use was 67. My velve-estimated exp_cov was 21 and my histogram-estimated
cov_cutoff was 16 The auto gave better statistics"
```

Print info on nucleotide coverage in raw reads

```
# Print the number of lines in one of the sequence files in the ~/quizzes/FINAL/ folder.
#cd quizzes/FINAL
expr $(cat Lmesenteroides1191_00for.fastq | wc -l)
```

```
## 13305600
```

```
# State the expected size of your genome
print("My expected genome size is 3.25 megabases")
```

```
## [1] "My expected genome size is 3.25 megabases"
```

Print info for splitting genome, if necessary

```
# If necessary, split the genome. If so, store the split file in your ~/velvet_1.2.10/data folde
r, show the number of lines in the file, and re-esimate nucleotide coverage. If it takes you mul
tiple tries you do not need to document all your error-tries; just show the time it worked. Make
sure to show all code for the time that got you between kmer coverage 20 and 30.
split -a 2 -l 1972176 Lmesenteroides1191_00for.fastq --additional-suffix=Lmesen_for.fastq
split -a 2 -l 1972176 Lmesenteroides1191_00rev.fastq --additional-suffix=Lmesen_rev.fastq
```

```
# print the number of lines in your split file
expr $(cat Lmesenteroides1191_00for.fastq | wc -l)
```

```
## 13305600
```

```
# Print the estimated nucleotide coverage from your split file
print("153")
```

```
## [1] "153"
```

run velvet

```
velvet_1.2.10/velveth lmesen 51,83,8 -fastq -shortPaired xaaLmesen_for.fastq xaaLmesen_rev.fastq

for i in {51..75..8}; do velvet_1.2.10/velvetg lmesen_$i -cov_cutoff auto -ins_length 450 -exp_c
ov auto -read_trkg yes; done;
```

Print table with assembly statistics

```
tail -n 1 lmesen_*/Log | sed ':a;N;$!ba;s/\n//g' | sed 's/lmesen/\n/g' | cut -f 1,5,10,12,14 -d
" " --output-delimiter '  ' | sed 's/,//g' | sed 's/Log//g' | sed 's/_//g' | sed 's/==>//g' | tr
-d '/' | sort -k1 -n
```

```
##
## 51  603  45527  266539  2132124
## 59  594  46322  298988  2137148
## 67  497  43615  197184  2098015
## 75  521  46338  299004  2143083
```

Create histogram; estimate cov_cutoff and exp_cov

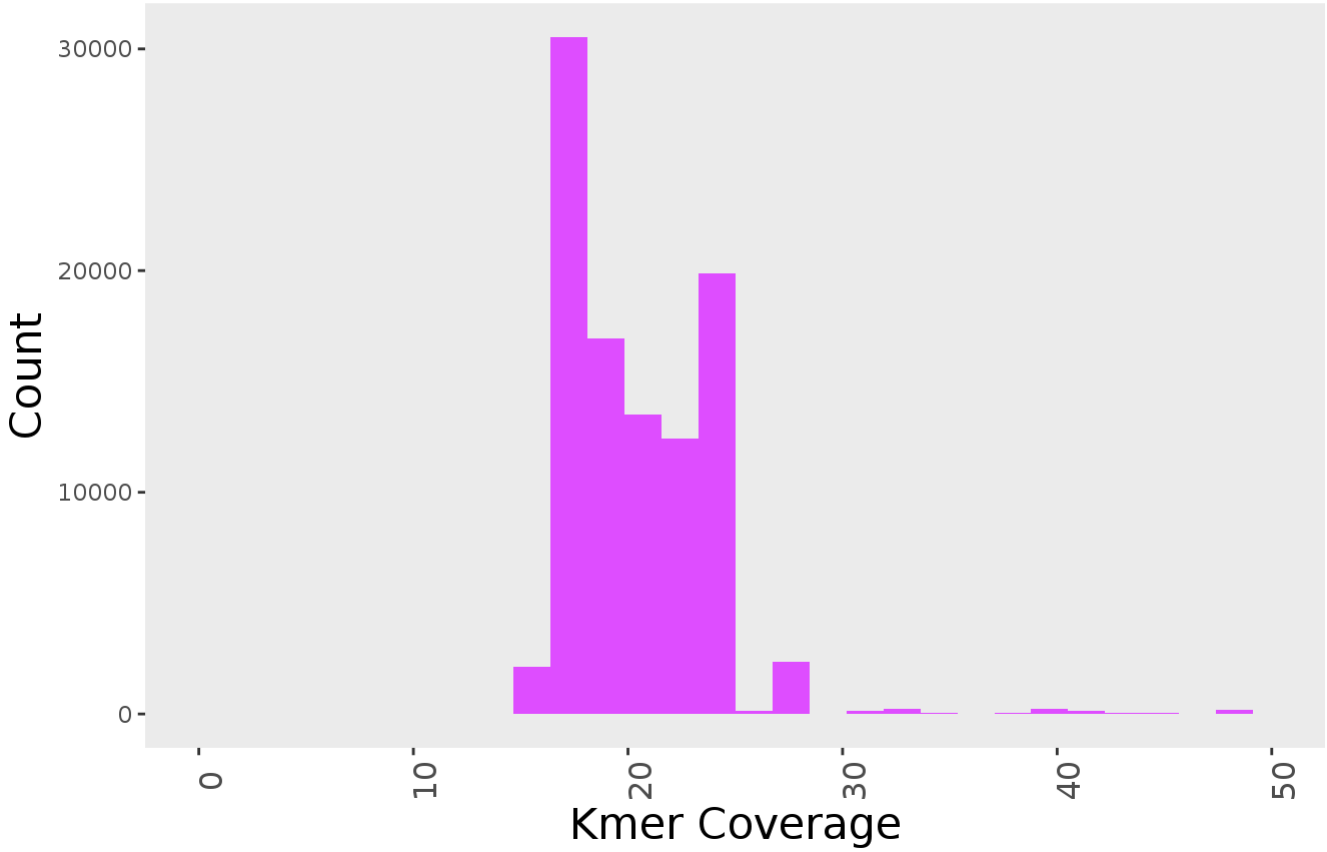
```
data_67 <- read.table('lmesen_67/stats.txt', header = T, sep = "\t", fill = T) %>% mutate(cum_lg
th = cumsum(lgth))
ggplot(data_67, aes(short1_cov,weight = lgth/short1_cov )) + geom_histogram(fill = "#de4dff") +
 xlim(0,50) + labs(title = "Histogram of kmers") +
  theme(title = element_text(size = 24), axis.title.x = element_text(size = 16), axis.title.y =
 element_text(size = 16), axis.text.x = element_text(size = 12, angle = 90), legend.position =
"none", panel.grid = element_blank()) +
  xlab("Kmer Coverage") +
  ylab("Count")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 229 rows containing non-finite values (stat_bin).
```

```
## Warning: Removed 2 rows containing missing values (geom_bar).
```

Histogram of kmers



Rerun velvetg

```
velvet_1.2.10/velvetg lmesen_67 -cov_cutoff 16 -ins_length 450 -exp_cov 21 -read_trkg yes
```

Print table comparing auto and manual

```
#tail -n 24 lmesen_67/Log | cut -f 3,13,15,24 -d $'\n' | cut -f 4,5,9,11,13 -d ' ' | sed 'N;s/
\n/ /' | cut -f 2,5,7,8,9 -d ' ' | sed 's/\,//g' | sed 's/16 /manual /g' | sed 's/ /\t/g'

tail -n 24 lmesen_67/Log | sed ':a;N;$!ba;s/\n//g' | sed 's/Final/\n/g' | grep -n 'graph' |cut -
f 1,9,11,13 -d " " --output-delimiter ' ' | sed 's/,/\t/g'  | sed 's/2: /auto /g' |  sed 's/3: /
manual /g'
```

```
## auto 46330    298996  2139876
## manual 43615  197184  2098015
```

Genome Analysis

Read in your amino acid fasta files (5 pts)

```
# 0.5 pts for changing each genome's names before and after, and showing it
# 1.5 pts for showing the length of the file you will submit matches the length of those files
# 1 pts for printing the length of the file you'll submit.

# Remember to include your HW9 genome as the fifth genome in this analysis
thermo_aa<- read.fasta("thermo_amino.faa")
head(names(thermo_aa), n = 6)
```

```
## [1] "fig|1308.1331.peg.426" "fig|1308.1331.peg.427" "fig|1308.1331.peg.428"
## [4] "fig|1308.1331.peg.429" "fig|1308.1331.peg.430" "fig|1308.1331.peg.431"
```

```
names(thermo_aa) <- gsub("fig", "thermo", names(thermo_aa))
head(names(thermo_aa), n = 6)
```

```
## [1] "thermo|1308.1331.peg.426" "thermo|1308.1331.peg.427"
## [3] "thermo|1308.1331.peg.428" "thermo|1308.1331.peg.429"
## [5] "thermo|1308.1331.peg.430" "thermo|1308.1331.peg.431"
```

```
lrham_aa<- read.fasta("lrham_amino.faa")
head(names(lrham_aa), n = 6)
```

```
## [1] "fig|47715.960.peg.724" "fig|47715.960.peg.725" "fig|47715.960.peg.726"
## [4] "fig|47715.960.peg.727" "fig|47715.960.peg.728" "fig|47715.960.peg.729"
```

```
names(lrham_aa) <- gsub("fig", "lrham", names(lrham_aa))
head(names(lrham_aa), n = 6)
```

```
## [1] "lrham|47715.960.peg.724" "lrham|47715.960.peg.725"
## [3] "lrham|47715.960.peg.726" "lrham|47715.960.peg.727"
## [5] "lrham|47715.960.peg.728" "lrham|47715.960.peg.729"
```

```
lplant_aa<- read.fasta("lplant_amino.faa")
head(names(lplant_aa), n = 6)
```

```
## [1] "fig|1590.2194.peg.591" "fig|1590.2194.peg.592" "fig|1590.2194.peg.593"
## [4] "fig|1590.2194.peg.594" "fig|1590.2194.peg.595" "fig|1590.2194.peg.596"
```

```
names(lplant_aa) <- gsub("fig", "lplant", names(lplant_aa))
head(names(lplant_aa), n = 6)
```

```
## [1] "lplant|1590.2194.peg.591" "lplant|1590.2194.peg.592"
## [3] "lplant|1590.2194.peg.593" "lplant|1590.2194.peg.594"
## [5] "lplant|1590.2194.peg.595" "lplant|1590.2194.peg.596"
```

```
lmesen_aa<- read.fasta("lmesen_amino.faa")
head(names(lmesen_aa), n = 6)
```

```
## [1] "fig|1245.234.peg.574" "fig|1245.234.peg.575" "fig|1245.234.peg.576"
## [4] "fig|1245.234.peg.577" "fig|1245.234.peg.578" "fig|1245.234.peg.579"
```

```
names(lmesen_aa) <- gsub("fig", "lmesen", names(lmesen_aa))
head(names(lmesen_aa), n = 6)
```

```
## [1] "lmesen|1245.234.peg.574" "lmesen|1245.234.peg.575"
## [3] "lmesen|1245.234.peg.576" "lmesen|1245.234.peg.577"
## [5] "lmesen|1245.234.peg.578" "lmesen|1245.234.peg.579"
```

```
hw9_aa <- read.fasta("hw9.faa")
head(names(hw9_aa), n = 6)
```

```
## [1] "fig|1598.1444.peg.479" "fig|1598.1444.peg.480" "fig|1598.1444.peg.481"
## [4] "fig|1598.1444.peg.482" "fig|1598.1444.peg.483" "fig|1598.1444.peg.484"
```

```
names(hw9_aa) <- gsub("fig", "hw9", names(hw9_aa))
head(names(hw9_aa), n = 6)
```

```
## [1] "hw9|1598.1444.peg.479" "hw9|1598.1444.peg.480" "hw9|1598.1444.peg.481"
## [4] "hw9|1598.1444.peg.482" "hw9|1598.1444.peg.483" "hw9|1598.1444.peg.484"
```

```
#write.fasta(thermo_aa, names(thermo_aa), file.out = "combined_aa.faa", open = "a")
#write.fasta(lrham_aa, names(lrham_aa), file.out = "combined_aa.faa", open = "a")
#write.fasta(lplant_aa, names(lplant_aa), file.out = "combined_aa.faa", open = "a")
#write.fasta(lmesen_aa, names(lmesen_aa), file.out = "combined_aa.faa", open = "a")
#write.fasta(hw9_aa, names(hw9_aa), file.out = "combined_aa.faa", open = "a")


combined_aa <- read.fasta("combined_aa.faa")
length(combined_aa) == length(thermo_aa) + length(lrham_aa) + length(lplant_aa) + length(lmesen_
aa) + length(hw9_aa)
```

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

```
length(combined_aa)
```

```
## [1] 11258
```

```
length(thermo_aa) + length(lrham_aa) + length(lplant_aa) + length(lmesen_aa) + length(hw9_aa)
```

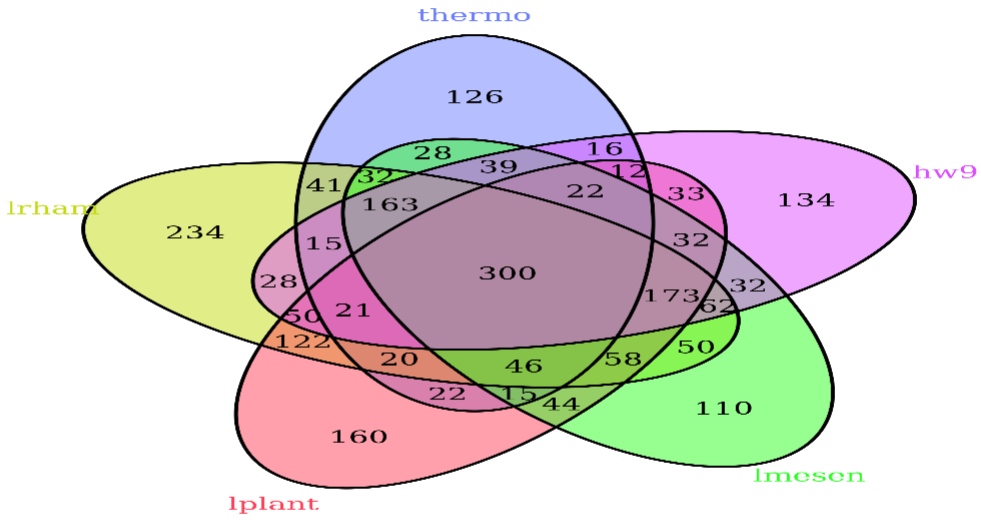
```
## [1] 11258
```

OrthoMCL work (12 pts)

```
# 3 pts for showing and completing all your work
# 3 pts for creating a correct and pretty VennDiagram
# 3 pts for listing the lengths and annotations for all the proteins that are present only in th
e union of the S. thermophilus, L. rhamnosus, and L. mesenteroides genomes.
# 3 pts for a summary paragraph that interprets 3 things about the Venn Diagram and/or the list
 you create.
ortho <- read.csv("orthoMCL.text", sep = "\t", fill = TRUE, row.names=NULL, header = FALSE)

vec1 <- ortho$V2[grep("thermo", ortho$V1)]
vec2 <- ortho$V2[grep("lrham", ortho$V1)]
vec3 <- ortho$V2[grep("lplant", ortho$V1)]
vec4 <- ortho$V2[grep("lmesen", ortho$V1)]
vec5 <- ortho$V2[grep("hw9", ortho$V1)]

colors <- c("#6b7fff", "#c3db0f", "#ff4059", "#2cff21", "#de4dff")

venn.diagram(x =list(vec1,vec2,vec3,vec4,vec5), category.names = c("thermo", "lrham", "lplant",
"lmesen", "hw9")  , filename = "keg_file", output = TRUE, imagetype = "png", scaled = FALSE, col
= "black", fill = colors, cat.col = colors, caat.cex = 2, margin = 0.15)
```

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

```
options(repr.plot.height=12, repr.plot.width=12)
library(png)
pp <- readPNG("keg_file")
plot.new()
rasterImage(pp, 0,0,1.1,1.1)
```


thermo
hw9
lrham
lmesen
lplant
126
28
16
39
12
41
32
22
33
134
163
234
15
32
300
28
173
32
21
62
50
122
20
46
58
50
22
15
44
110
160

```
venn_groups <- get.venn.partitions(x = list(thermo = vec1, lrham = vec2, lplant = vec3, lmesen =
vec4, hw9 = vec5))
all_vec <- unname(unlist(venn_groups[22,]$..values..))

sub_vec <- c()
for (i in 1:length(all_vec)){
  sub_vec <- c(sub_vec, which(ortho$V2 == all_vec[i]))

}
gen_names <- ortho[sub_vec,1]
#ortho[sub_vec,]


gen_names <- gsub("thermo", "fig", gen_names)
gen_names <- gsub("lrham", "fig", gen_names)
gen_names <- gsub("lplant", "fig", gen_names)
gen_names <- gsub("lmesen", "fig", gen_names)
gen_names <- gsub("hw9", "fig", gen_names)




annotation <- read.csv("annotation.txt", sep = "\t", fill = TRUE, header = TRUE)
annotation <- rbind(annotation, read.csv("1245.234.txt", sep = "\t", fill = TRUE, header = TRUE
), read.csv("47715.960.txt", sep = "\t", fill = TRUE, header = TRUE))

sub_vec <- c()
for (i in 1:length(gen_names)){
  sub_vec <- c(sub_vec, which(annotation$feature_id == gen_names[i]))

}

final_names <- annotation[sub_vec,]
final_names <- final_names[!is.na(final_names$contig_id),]
head(final_names)
```

```
##                                 contig_id             feature_id type
## 4883   NODE_1_length_106374_cov_23.762461  fig|47715.960.peg.737  peg
## 4884   NODE_1_length_106374_cov_23.762461  fig|47715.960.peg.738  peg
## 5989   NODE_3_length_127423_cov_26.342937 fig|47715.960.peg.1770  peg
## 2522   NODE_25_length_43619_cov_16.421284   fig|1245.234.peg.731  peg
## 4926   NODE_1_length_106374_cov_23.762461  fig|47715.960.peg.780  peg
## 2861 NODE_323_length_197184_cov_16.904369  fig|1245.234.peg.1066  peg
##                                                location  start   stop strand
## 4883      NODE_1_length_106374_cov_23.762461_9950_10348   9950  10348      +
## 4884     NODE_1_length_106374_cov_23.762461_10352_10747  10352  10747      +
## 5989     NODE_3_length_127423_cov_26.342937_57960_58367  57960  58367      +
## 2522     NODE_25_length_43619_cov_16.421284_36107_36514  36107  36514      +
## 4926     NODE_1_length_106374_cov_23.762461_58482_59360  58482  59360      +
## 2861 NODE_323_length_197184_cov_16.904369_123491_124381 123491 124381      +
##                                          function. aliases figfam
## 4883                Acetyltransferase, GNAT family      NA     NA
## 4884                Acetyltransferase, GNAT family      NA     NA
## 5989                Acetyltransferase, GNAT family      NA     NA
## 2522                conserved hypothetical protein      NA     NA
## 4926 Uncharacterized UPF0750 membrane protein YpjC      NA     NA
## 2861 Uncharacterized UPF0750 membrane protein YpjC      NA     NA
##      evidence_codes
## 4883
## 4884
## 5989
## 2522
## 4926
## 2861
##
nucleotide_sequence
## 4883
gtgtcgtacaccatcaaagtaaacgcacccttaacggttcaacaagttcatgatctttatcagcaaacgcattttgacaagcccatcgctgatgct
gcccgtctacaagtgatgattgatgaaacccaactggtcttgagtgtctgggacgatgagcatctcattggctttgcacgctgtctgaccgacttt
gagtactgttgctatctgagtgacattttaattctgcccgcttatgaaggccaccaaattgggcggcaattgatcgcgactttacaagcttacatc
ggcccaagggttactttatcattgcgcgcggctgacagcgcagtaggtttctatgaacgcattaatttgccacacgctgacaacatgtttcgtatc
ggtcgggagggataa
## 4884
ttgttacatattgaaaataatcagccgatctcggcctcgcagtttattgccgttttagatgcatccggtattcaccggccgactgaagatcgtgcg
cgcatgcagcgaatgcttaataacgcaaatgtgttgctcaccgcatgggatgatcaccagttaatcggcgttttacgtggtgtgtctgacaagtcc
tattgcacgtttgtttccgagctagctgttatcaaaagccaccagcatcaaggtgtgggcaaggcactattgcaaacactgcataccatccaggga
cccaacatttcaatcatgttgctatcggccccagcagccatgcagttttatcccaaagtcgactttaaacctgttccgacggctttcaaagttcaa
cggcagttttaa
## 5989
atgattaaaattgatcagcgtcagttaaagcaggcagatgtgcttgccctttatcaagcggtcggttggaacatgtacacgcgagatccgaagaaa
cttgagcgggcgatagcgcaatcgttaagcgtcctgggggcttatgatggcgatcggttagttggtttgattcgggcagttggcgatggcgagacg
attttatttattcaggatttactcgttttgcccagttatcaacggcaggggatcgggcggcaattggtaaacgcattagtggatcagtttccacag
gttcgtcagcgggtactttaaccgatgatcagccccaaactcgcgccttttacgaaaatattggctttgtgcaaagtagtaaagttggtgtgatt
gctttttatcaggacttggcgtaa
## 2522
atgatcaattatcaaatcaatcaaacaatcgcgaaaactgatttaaccaaattgtacaatagtgtcggttggtttgcctatactaacactcgtgta
aaccttatggctgctgttgcaaattcgttaatggttgtcagtgcgtgggcagataatcaattagtcggattagttcgcattgtgggggatggagag
acaattatttttatacaagatattctagttgacccgaagtttcaaaatcaacatattggcacagagctaatgaatcgtgttttaagtcagtatcca
gcagttaggcaaaaagtattattgactgaagaagcgcctgatgtcaggcatttttatgaaaaatttggtttcaaatctgctgatcaagggacgctt
```

```
gtcgtttttataaaaatttttaa
## 4926             atgtcacgtaacacgcgaatcggactggatcttttagtgatcactttgggctgtgcgctttacggctttggactcg
tgtatatcaatattgccaatcacctagcagaaggtggcgtcaccgggatcactctgctgattcgttactggtggggtctggatccggcttatagca
cggttttgttgaatattcctttgctgattgtgggctacaaatttttgggcaagcgtgctttagcctataccatctatggaactttgatgttatctg
cttggttgtggatttggcaacgggtgcccttaagcattgatattcaccatgacttgtttatatccggggttttagccgggctttttggcggctttg
gtagtgggattatttatcgacatggcggaaccactggaggtaccgatgtcgttgcccgcatccttgagcagcaaaccggcgtgccaatggggcgaa
cgctgctcatattcgatgccattgtcttgaccgtttccctgacatacttaaacattgagctcatgatgtacacgctgcttggcgcctacgttttct
cgcgcatcgtcaactttacgcttgatggtgcctatgctgccaagggtgtcttagtcgtatcggaccacagccaggcaattgccacggcgattatgg
acgagcttgaacgcggaacgacctttttgcatgcagaaggcggctttgcccatgaccgcaagcaagtcgtttacgccgtggtcgcttccagcgaaa
ttgcccacacaaaacggctgattgaagccattgatccgcgtgcttttattagcattctggatgtccatgaagcactgggtgaaggattcacttatc
aaaagaagcgccgacgccttttattcggccattaa
## 2861 atgaatgtaacaaatttagaacgcatttctgtgcgtgatattggtatgattgcattgggaacagccctttatggttggggtttaatca
acattaatattccaaatcaactggcagaaggtggcgtttctggtatcacgttgattttacgcgcgcttttttggttggaacccagcttataccacac
tattattgaatattccgcttgtttttattggctatcgcgtactcggtagacgatctttaatttatacaatttggggtatttcatcgctttcatttt
ggttgtggttatggcaagttgtcccgacaccacccgcacttgatcacgacatgctaatcgcaggtttgctggcaggatttatttctggtttaggtt
taggcattgtctttcgatttggcggtacatcaggtggtacagacatcgttgcaaaaattacagaacaaaaactaggcatccaaattggtcgaacaa
tgtttgcgcttgatgcagttgttctgattatttctttaatttatatcgacattgtacacatgatgtatactttgatagcgtcatttgtatttgcac
aagttgttaattttacacagcaaggtgctgcatactcagcaagatcatttatgattttacacagtatcctgaagaaatatcccatgcgattatgt
cagaactcgatcgcggcaccagtctgttaaaagcagaaggtggctactcacacattgatcagcgcgtcgtttatgctgtcgtcgatccttcggaag
tcaacatggtcaggcacatcattaatcaaattgatccaaaagcatttgtttccgtatttgacacacaagaacaattaggcgaaggcttttcctact
tgcgccccaaaaaatcaatatttaaattcaaataa
##
aa_sequence
## 4883
MSYTIKVNAPLTVQQVHDLYQQTHFDKPIADAARLQVMIDETQLVLSVWDDEHLIGFARCLTDFEYCCYLSDILILPAYEGHQIGRQLIATLQAYI
GPRVTLSLRAADSAVGFYERINLPHADNMFRIGREG
## 4884
MLHIENNQPISASQFIAVLDASGIHRPTEDRARMQRMLNNANVLLTAWDDHQLIGVLRGVSDKSYCTFVSELAVIKSHQHQGVGKALLQTLHTIQG
PNISIMLLSAPAAMQFYPKVDFKPVPTAFKVQRQF
## 5989
MIKIDQRQLKQADVLALYQAVGWNMYTRDPKKLERAIAQSLSVLGAYDGDRLVGLIRAVGDGETILFIQDLLVLPSYQRQGIGRQLVNALVDQFPQ
VRQRVLLTDDQPQTRAFYENIGFVQSSKVGVIAFYQDLA
## 2522
MINYQINQTIAKTDLTKLYNSVGWFAYTNTRVNLMAAVANSLMVVSAWADNQLVGLVRIVGDGETIIFIQDILVDPKFQNQHIGTELMNRVLSQYP
AVRQKVLLTEEAPDVRHFYEKFGFKSADQGTLVVFYKNF
## 4926     MSRNTRIGLDLLVITLGCALYGFGLVYINIANHLAEGGVTGITLLIRYWWGLDPAYSTVLLNIPLLIVGYKFLGKRALAYTIYG
TLMLSAWLWIWQRVPLSIDIHHDLFISGVLAGLFGGFGSGIIYRHGGTTGGTDVVARILEQQTGVPMGRTLLIFDAIVLTVSLTYLNIELMMYTLL
GAYVFSRIVNFTLDGAYAAKGVLVVSDHSQAIATAIMDELERGTTFLHAEGGFAHDRKQVVYAVVASSEIAHTKRLIEAIDPRAFISILDVHEALG
EGFTYQKKRRRLLFGH
## 2861 MNVTNLERISVRDIGMIALGTALYGWGLININIPNQLAEGGVSGITLILRALFGWNPAYTTLLLNIPLVFIGYRVLGRRSLIYTIWGI
SSLSFWLWLWQVVPTPPALDHDMLIAGLLAGFISGLGLGIVFRFGGTSGGTDIVAKITEQKLGIQIGRTMFALDAVVLIISLIYIDIVHMMYTLIA
SFVFAQVVNFTQQGAAYSARSFMIFTQYPEEISHAIMSELDRGTSLLKAEGGYSHIDQRVVYAVVDPSEVNMVRHIINQIDPKAFVSVFDTQEQLG
EGFSYLRPKKSIFKFK
```

```
print("The intersection of all the genomes have the highest count, so each genome are highly rel
ated to each other. Irham has the most unique amount of proteins. Lmensen has the least amount o
f unique proteins.")
```

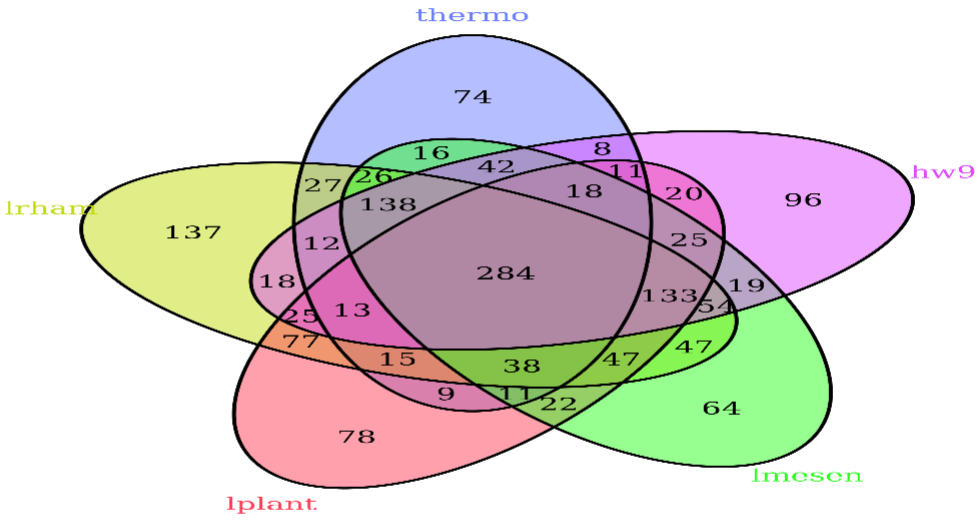
```
## [1] "The intersection of all the genomes have the highest count, so each genome are highly re
lated to each other. Irham has the most unique amount of proteins. Lmensen has the least amount
of unique proteins."
```

KEGG work (12 pts)

```
# 3 pts for showing and completing all your work
# 3 pts for creating a correct and pretty VennDiagram
# 3 pts for the table resulting from differential abundance testing (remember: eyeball test, pos
sibly supported by a 'ratio' column - whatever is useful. No formal test here).
# 3 pts for a summary paragraph that interprets 3 things about the Venn Diagram and/or the list
 you create.
keg <- read.csv("user_ko.txt", sep = "\t", fill = TRUE, row.names=NULL, header = FALSE)

vec1 <- keg$V2[grep("thermo", keg$V1)]
vec2 <- keg$V2[grep("lrham", keg$V1)]
vec3 <- keg$V2[grep("lplant", keg$V1)]
vec4 <- keg$V2[grep("lmesen", keg$V1)]
vec5 <- keg$V2[grep("hw9", keg$V1)]

colors <- c("#6b7fff", "#c3db0f", "#ff4059", "#2cff21", "#de4dff")

venn.diagram(x =list(vec1,vec2,vec3,vec4,vec5), category.names = c("thermo", "lrham", "lplant",
"lmesen", "hw9")  , filename = "keg_file", output = TRUE, imagetype = "png", scaled = FALSE, col
= "black", fill = colors, cat.col = colors, caat.cex = 2, margin = 0.15)
```

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

```
options(repr.plot.height=12, repr.plot.width=12)
library(png)
pp <- readPNG("keg_file")
plot.new()
rasterImage(pp, 0,0,1.1,1.1)
```




```
print("The intersection of all the genomes have the highest count, so each genome are highly rel
ated to each other. Irham has the most unique amount of proteins. Lmensen has the least amount o
f unique proteins.")
```

```
## [1] "The intersection of all the genomes have the highest count, so each genome are highly re
lated to each other. Irham has the most unique amount of proteins. Lmensen has the least amount
of unique proteins."
```