K-mer counting tool instructions

Gabe Mednick

8/4/2021

Dear colleague,
Thanks for sharing your concern about the k-mer frequencies in your experiments. I had a chance to look at the FASTA files and, as you suspected, the k-mer counts are not evenly distributed across experiments.

This document contains a short analysis of the `challenge 1 FASTA files (Exp1-4)` and provides instructions on how to run the accompanying `kmer_counting_tool.R` script from the command line. I have the k-mer length set to 4, but you can change it to any value within the range of your sequence length when using the counting script.

Import the FASTA file with the Biostrings package

The nucleotide sequence will be imported as a DNAStringSet object but I will convert it into a data frame and slice it into k-mers.

```
## DNAStringSet object of length 1:
##     width seq                         names
## [1] 10000 GTATTTACAGCAA...TGTCTTTTATGAG
```

Table of K-mer counts by Experiment

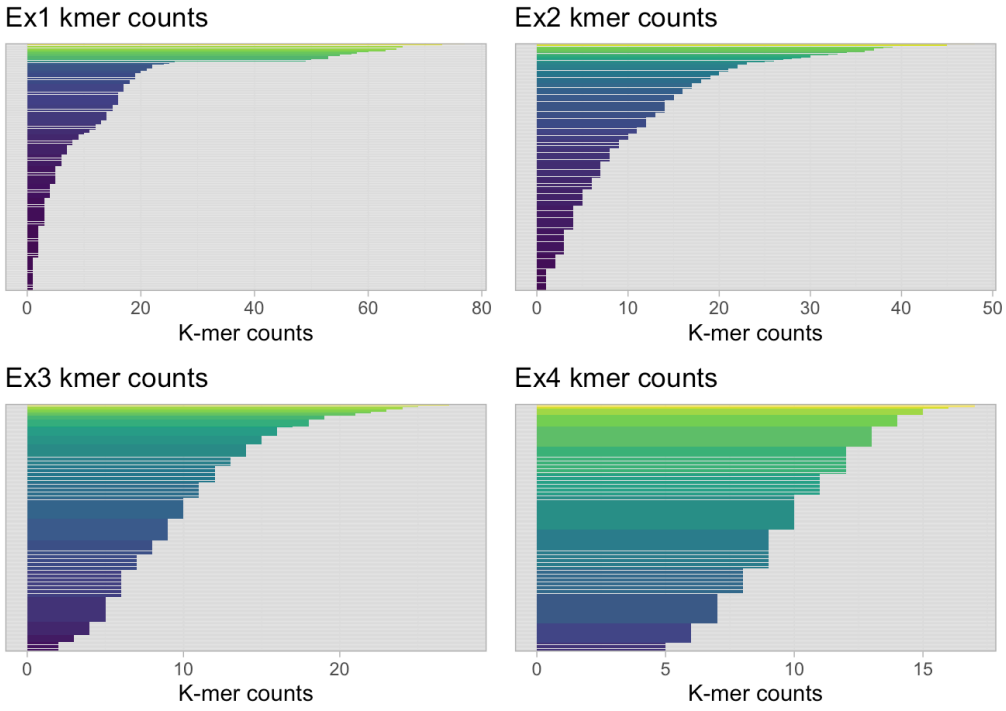
This table is similar to the tab separated output file that is produced from the `kmer_counting_tool.R`.

Kmer counts by experiment

kmer	standard_nucs	Exp1_counts	Exp2_counts	Exp3_counts	Exp4_counts
ATAA	TRUE	77	45	22	10
TAAT	TRUE	73	37	19	8
ATAT	TRUE	66	38	24	9
TATT	TRUE	66	28	27	11
AAAT	TRUE	65	30	22	14
AATT	TRUE	65	36	15	11
TTTT	TRUE	63	29	28	9
TAAA	TRUE	60	30	24	9

It may be helpful to visualize the k-mer count distributions to look for similarities and differences between the experiments.

Bar plots of count distributions for each experiment



Notably, experiments 1 and 2 are right skewed and experiments 3 and 4 are progressively less so.

Summary table

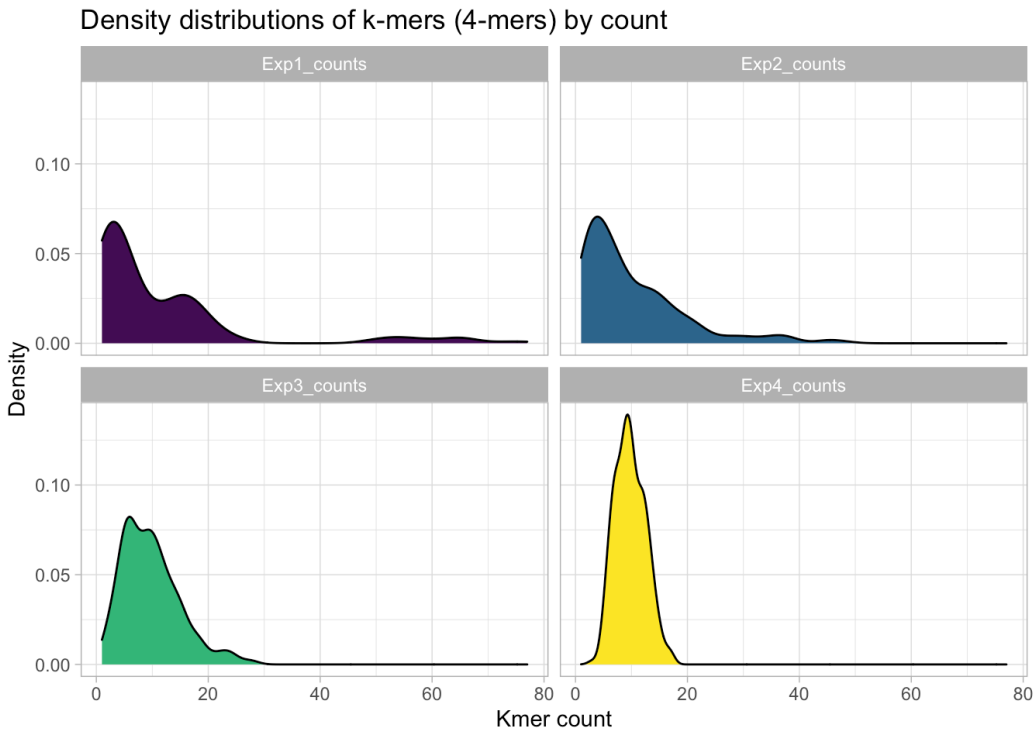
The `max count` variation is worth exploring further.

K-mer counts summary table

experiment	median count	mean count	min count	max count
Exp1_counts	5	11	1	77
Exp2_counts	7	10	1	48
Exp3_counts	9	10	1	28
Exp4_counts	10	10	3	17

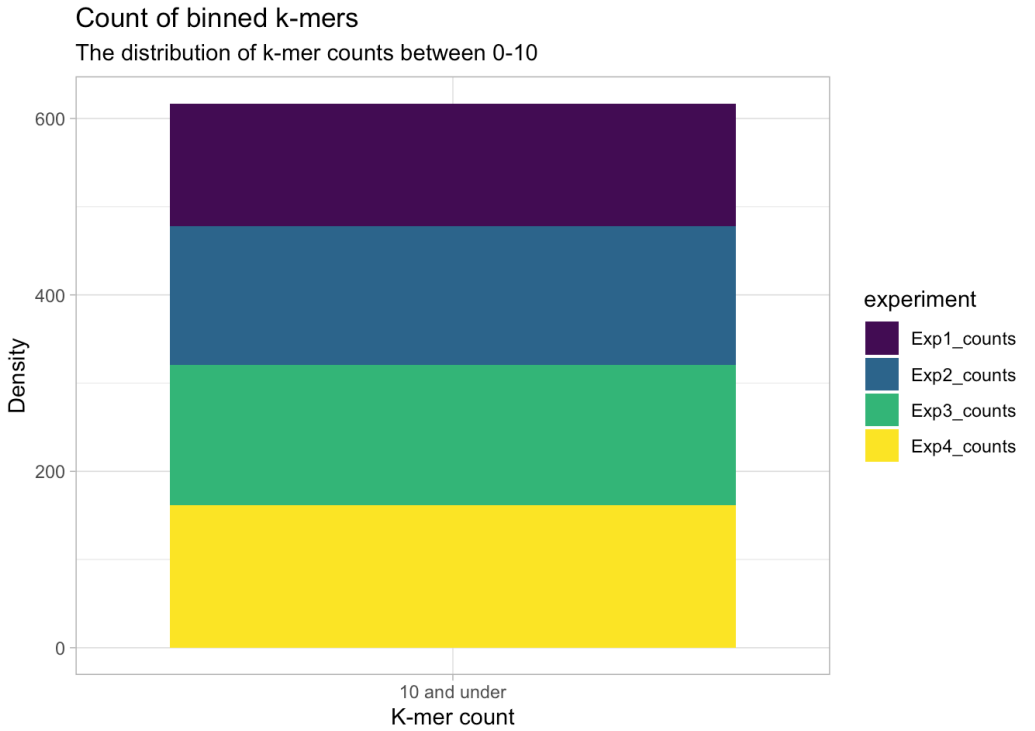
Density plots

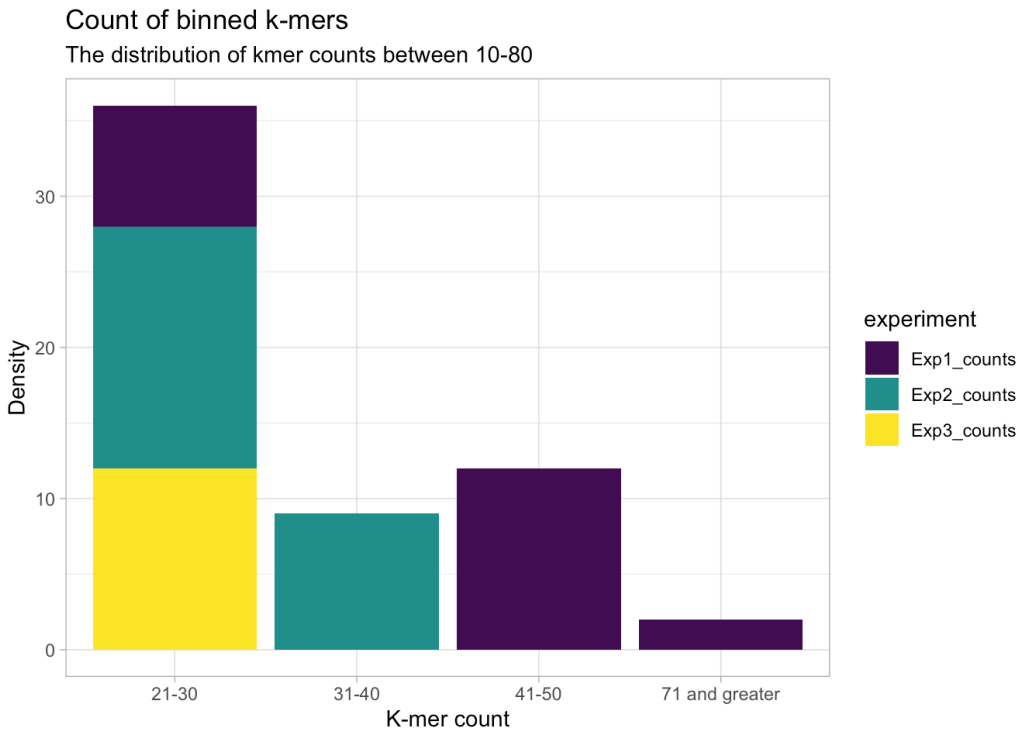
We can get a better feeling for the distributions with density plots.



Binning the kmer counts for a more general pattern

In the next two plots, the k-mer counts are grouped into 8 categories (10-80 k-mer counts by 10 and an other category). For all experiments, most of the 256 possible k-mers appear less than 10 times. So as not to dwarf the k-mer counts that appear more than ten times, it's better shown separately.





We see that experiment 2 dominates k-mer counts in the 30-40 range and experiment 1 dominates in the greater than 40 range.

I hope these plots help you narrow down the culprit behind your experimental variation. Good luck and let me know if you have any questions regarding the command line k-mer counting script.

Instructions: k-mer counting script for the command line

To help you check for imbalanced k-mer distributions in future experiments, I designed a k-mer counting analysis script that works from the command line. The k-mer counter let's you input a FASTA file and k-mer length, and returns a tab separated file of k-mer counts ordered by frequency. The program also returns messages about the analysis including:
- Input sequence length
- K-mer length
- Which nonstandard nucleotides the program can identify
- Whether the k-mer length is acceptable for the sequence range
- The top 10 k-mers by count
- Whether the given sequence contains standard or nonstandard nucleotides. The nonstandard nucleotide warning can be tested with the following file: takehome/challenge1/nonstandard_nucs.fasta

To use the k-mer counting tool:
1. Download the directory I sent you and open it in the command line
2. Change the permissions for kmer_counter_tool.R script to make it executable on your machine (chmod +x kmer_counter_tool.R)
3. Then run the following incantation in your command line (with custom input and output file names):
Rscript kmer_counter_tool.R 'input_file' kmer-length –output_file 'output_file'
e.g.,

```
bio-rad [main]$ Rscript kmer_counting_tool.R 'takehome/challenge1/experiment1.fasta' 4 --output_file 'output-kmer-4.tsv'
```

4. See the image below for the expected output in the command line (Note: I am working on a mac).

When running the script, a new output file (tab separated format) with k-mers ordered by frequency is generated.

```
[[1]]
10000-letter DNAString object
seq: GTATTTACAGCAAAATTATATATAAAATGGGCAATT...ATTGACAGTATTTACTGCCATTTTGTCTTTTATGAG

[1] "The input sequence is 10000 bp's in length"
[1] "The kmer length is 4"
[1] "Good choice! The kmer length is within the sequence range."
[1] "Non-standard nucleotides include [bdefhijklmnopqrsuvwxyzBDEFHIJKLMNOPQRSUVWXYZ]"
# A tibble: 218 × 2
   kmer  length
   <chr>  <int>
 1 ATAA      77
 2 TAAT      73
 3 ATAT      66
 4 TATT      66
 5 AAAT      65
 6 AATT      65
 7 TTTT      63
 8 TAAA      60
 9 ATTT      58
10 TTAT      57
# … with 208 more rows
[1] "Great news: Your fasta sequence contains A, C, G, T"
```