More graphs and muti-part figures in R

Joann Mudge

January 25, 2021

1 Data

We will graph statistics from the assemblathon assemblies. You should alread have the data in your ggplot2 folder.

- Navigate to your ggplot2 folder.
- Don't forget to "source activate visualization".
- Open R.
- Load the ggplot2 library.

```
library(ggplot2)
```

2 Using for loops in R

For these graphs, we will use the scaffold lengths from three of the Assemblathon2 fish assemblies. First, we need to load in the data. Because there is a data file for each assembly, we need to load multiple files, which can be easily done with a for loop. For loops run a set of commands for each input you give it. Here, for each file, we will read it in and add it onto a data frame.

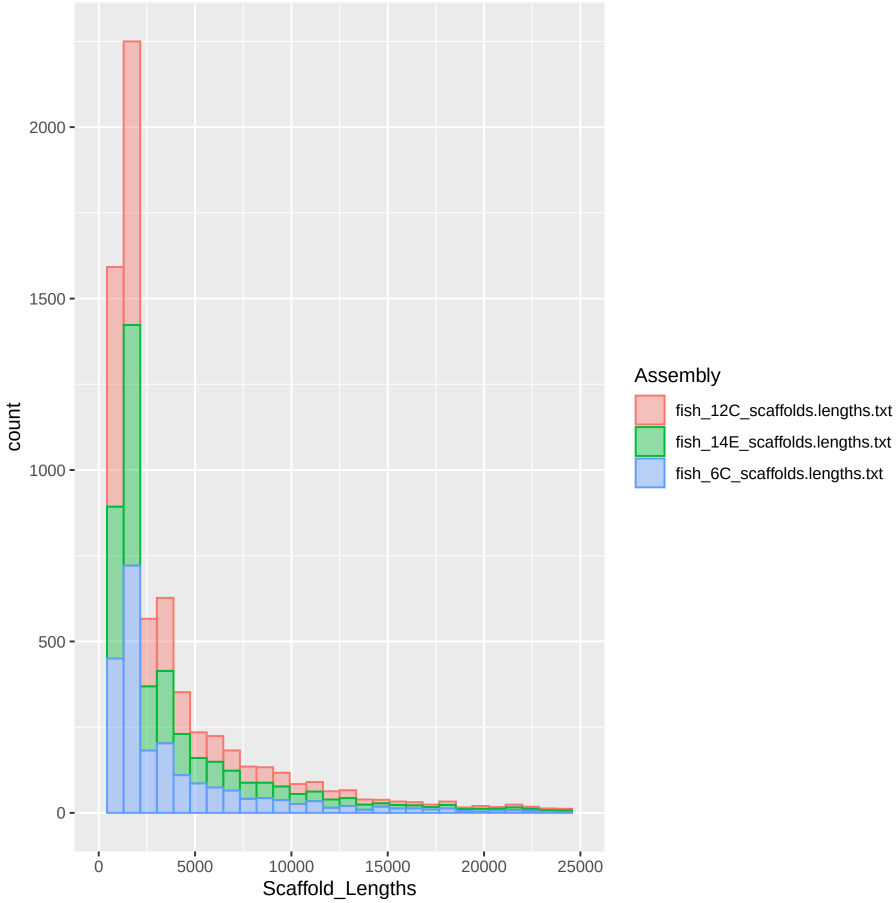
```
# Get all the filenames that end in lengths.txt
file.names = dir(".", pattern="*lengths.txt")

# Set up a variable that we'll dump all the data into
combo=""

# Read each file in and add it to the data frame
for(i in 1:length(file.names)) {
        file = read.table(file.names[i],header=FALSE)
        # Add a column with the assembly
        # Actually it is the name of the assembly file but we'll fix it later
        file$Assembly = file.names[i]
        combo = rbind(combo,file)
}

# Take a look
head(combo)

##         V1                       Assembly
## 1
## 2 2293760 fish_12C_scaffolds.lengths.txt
## 3 2293760 fish_12C_scaffolds.lengths.txt
## 4 2293760 fish_12C_scaffolds.lengths.txt
## 5 2293760 fish_12C_scaffolds.lengths.txt
## 6 2293760 fish_12C_scaffolds.lengths.txt

# What are the dimensions of the data frame?
dim(combo)

## [1] 8786    2

# Remove the first row because it is blank
combo=combo[-1,]
head(combo)

##         V1                       Assembly
## 2 2293760 fish_12C_scaffolds.lengths.txt
## 3 2293760 fish_12C_scaffolds.lengths.txt
## 4 2293760 fish_12C_scaffolds.lengths.txt
## 5 2293760 fish_12C_scaffolds.lengths.txt
## 6 2293760 fish_12C_scaffolds.lengths.txt
## 7 2293760 fish_12C_scaffolds.lengths.txt

# Strip the prefix and suffix so we are just left with the Assembly name
combo$Assembly = sub("fix.","",combo$Assembly)
combo$Assembly = sub("_scaffolds.fa.lengths.txt","",combo$Assembly)
head(combo)

##         V1                       Assembly
## 2 2293760 fish_12C_scaffolds.lengths.txt
## 3 2293760 fish_12C_scaffolds.lengths.txt
## 4 2293760 fish_12C_scaffolds.lengths.txt
## 5 2293760 fish_12C_scaffolds.lengths.txt
## 6 2293760 fish_12C_scaffolds.lengths.txt
## 7 2293760 fish_12C_scaffolds.lengths.txt
```

```
# Add column names
names(combo) = c("Scaffold_Lengths","Assembly")

# We need to make sure the scaffold lengths are numeric
sapply(combo,class)

## Scaffold_Lengths         Assembly
##      "character"      "character"

combo$Scaffold_Lengths = as.numeric(combo$Scaffold_Lengths)
sapply(combo,class)

## Scaffold_Lengths         Assembly
##        "numeric"      "character"
```

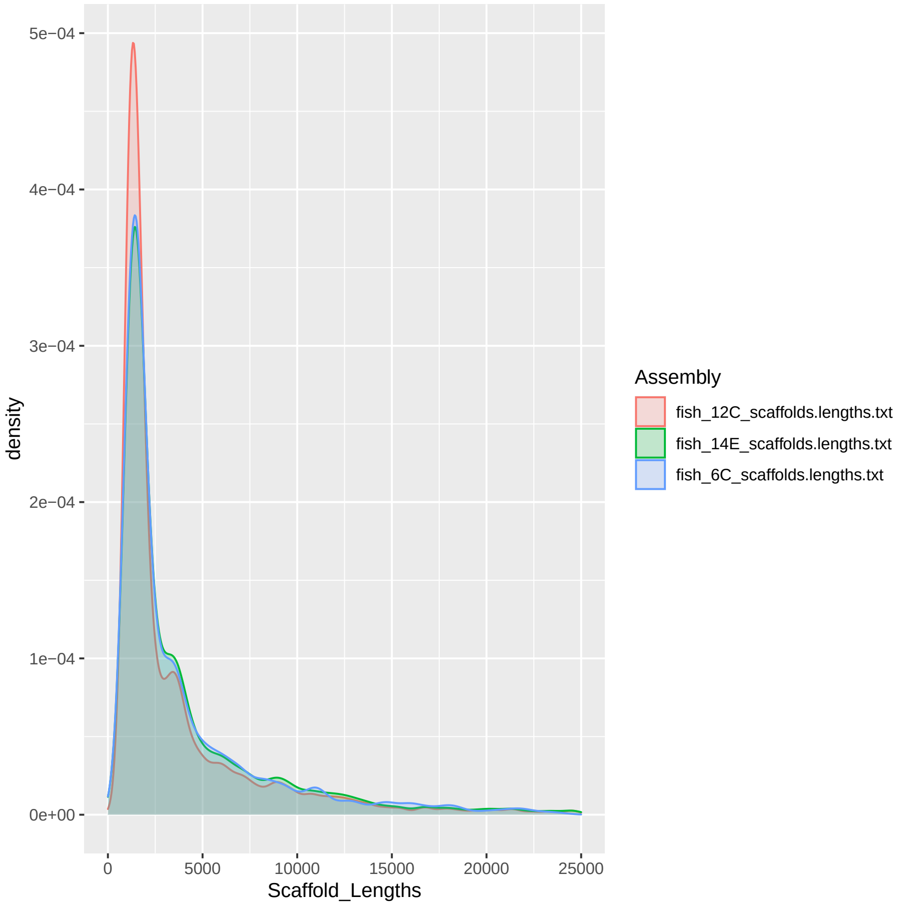
3 Histogram and density plots

ggplot2 allows us to do histogram and density plots. Density plots are just like histograms except instead of binning data, they visualize data over continuous intervals, essentially making it a "smooth" histogram. For data, we will use the scaffold lengths, focusing in on the smaller contigs (<=25,000 nt).

```
pdf("1.histo_dens_plots.pdf")

# Histogram
ggplot(combo,aes(x=Scaffold_Lengths,fill=Assembly,color=Assembly))+
        geom_histogram(alpha=0.4) + xlim(0,25000)

## ‘stat_bin()‘ using ‘bins = 30‘.  Pick better value with ‘binwidth‘.
## Warning:  Removed 1747 rows containing non-finite values (stat_bin).
## Warning:  Removed 6 rows containing missing values (geom_bar).

# Density plot
ggplot(combo,aes(x=Scaffold_Lengths,fill=Assembly,color=Assembly))+
        geom_density(alpha=0.2) + xlim(0,25000)

## Warning:  Removed 1747 rows containing non-finite values (stat_density).

dev.off()

## pdf
##   2
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## Warning: Removed 1747 rows containing non-finite values (stat_bin).
## Warning: Removed 6 rows containing missing values (geom_bar).
```



```
## Warning: Removed 1747 rows containing non-finite values (stat_density).
```



4 Publication quality and multi-part figures

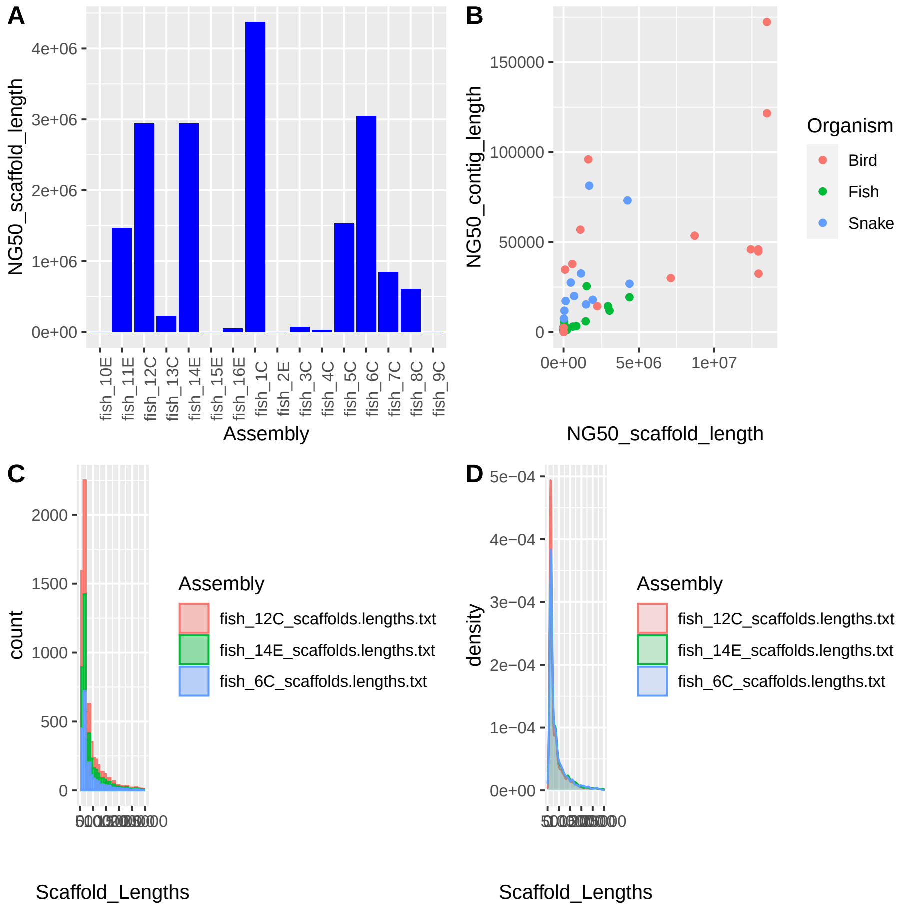
Cowplot is specifically designed for getting figures ready for publication and makes it easy to generate multi-part figures. We'll use four graphs that you should be familiar with. Let's get the assemblathon data for fish, bird and snake and merge it together. You have done this before.

```
fishstats = read.table("fish.stats.txt", header=TRUE)
snakestats = read.table("snake.stats.txt", header=TRUE)
birdstats = read.table("bird.stats.txt", header=TRUE)

# Before we merge them, let's get a column called organism into each data frame.
fishstats$Organism = "Fish"
snakestats$Organism = "Snake"
birdstats$Organism = "Bird"
```

```
# Unforunately, rbind won’t bind datasets with a different number of columns
        # (fishstats has two extra columns)
allstats = rbind(fishstats, snakestats, birdstats)

## Error in rbind(deparse.level, ...):  numbers of columns of arguments
do not match

# Remove columns 8 and 9 from fishstats while you bind it
allstats = rbind(fishstats[, -c(8:9)], snakestats, birdstats)
```

We will make 4 graphs and put them into variables. Then we’ll put the variables with the graphs into a multi-part figure.

```
library(cowplot)

figA = ggplot(fishstats,
        aes(x=Assembly, y=NG50_scaffold_length) ) +
        geom_bar(stat="identity", fill="blue") +
        theme(axis.text.x=element_text(angle=90))

figB = ggplot(allstats, aes(x=NG50_scaffold_length,y=NG50_contig_length,color=Organism)) +
        geom_point()

figC = ggplot(combo,aes(x=Scaffold_Lengths,fill=Assembly,color=Assembly))+
        geom_histogram(alpha=0.4) + xlim(0,25000)

figD = ggplot(combo,aes(x=Scaffold_Lengths,fill=Assembly,color=Assembly))+
        geom_density(alpha=0.2) + xlim(0,25000)

pdf("2.4x4pub.pdf")

# Plot 2x2
plot_grid(figA, figB, figC, figD, labels = c("A", "B", "C", "D"), ncol = 2,
        align="h")

## ‘stat_bin()‘ using ‘bins = 30‘.  Pick better value with ‘binwidth‘.
## Warning:  Removed 1747 rows containing non-finite values (stat_bin).
## Warning:  Removed 6 rows containing missing values (geom_bar).
## Warning:  Removed 1747 rows containing non-finite values (stat_density).

dev.off()

## pdf
##   2
## `stat_bin()` using `bins = 30`.  Pick better value with `binwidth`.
## Warning:  Removed 1747 rows containing non-finite values (stat_bin).
## Warning:  Removed 6 rows containing missing values (geom_bar).
## Warning:  Removed 1747 rows containing non-finite values (stat_density).
```

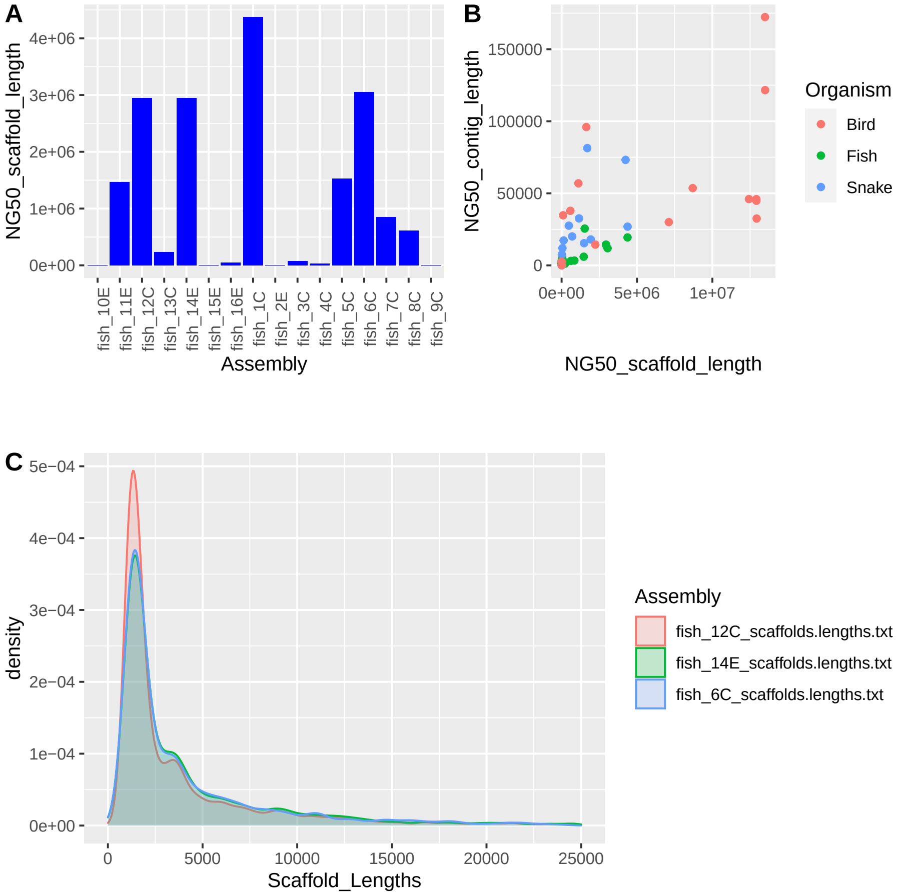


Easy, right? Unfortunately the histogram and the density plot are super squished. Let's just keep the density plot on the bottom row and make it go across the whole page. We'll do a nested plot grid. We'll put a 1x2 plot grid (the top row in the previous graph) into a 2x1 grid.

```
pdf("3.nestedpub.pdf")

# Plot top row
top=plot_grid(figA, figB, labels = c("A", "B"), ncol = 2,
        align="h")

# Plot the nested grid
```

```
plot_grid(top, figD, labels = c("", "C"), ncol = 1,
        align="h")

## Warning:  Removed 1747 rows containing non-finite values (stat_density).

dev.off()

## pdf
##   2
```

```
## Warning:  Removed 1747 rows containing non-finite values (stat_density).
```



- Homework (or if you get done early)
    - Go through the cowplot introductory vignette
        - (https://cran.r-project.org/web/packages/cowplot/vignettes/introduction.html)
    - Use ggplot2 to plot other types of graphs and generate a multi-part figure
        - (http://r-statistics.co/Top50-Ggplot2-Visualizations-MasterList-R-Code.html)