Imagine you want to make a [Manhattan-style](https://en.wikipedia.org/wiki/Manhattan_plot) plot or anything else where you want a series of intervals laid out on one axis after one another. If it’s actually a Manhattan plot you may have a friendly R package that does it for you, but here is how to cobble the plot together ourselves with *ggplot2*.

We start by making some fake data. Here, we have three contigs (this could be your chromosomes, your genomic intervals or whatever) divided into one, two and three windows, respectively. Each window has a value that we’ll put on the y-axis.

library(dplyr)

library(ggplot2)

data <- data\_frame(contig = c("a", "a", "a", "b", "b", "c"),

start = c(0, 500, 1000, 0, 500, 0),

end = c(500, 1000, 1500, 500, 1000, 200),

value = c(0.5, 0.2, 0.4, 0.5, 0.3, 0.1))

We will need to know how long each contig is. In this case, if we assume that the windows cover the whole thing, we can get this from the data. If not, say if the windows don’t go up to the end of the chromosome, we will have to get this data from elsewhere (often some genome assembly metadata). This is also where we can decide in what order we want the contigs.

contig\_lengths <- summarise(group\_by(data, contig), length = max(end))

Now, we need to transform the coordinates on each contig to coordinates on our new axis, where we lay the contings after one another. What we need to do is to add an offset to each point, where the offset is the sum of the lengths of the contigs we’ve layed down before this one. We make a function that takes three arguments: two vectors containing the contig of each point and the position of each point, and also the table of lengths we just made.

flatten\_coordinates <- function(contig, coord, contig\_lengths) {

coord\_flat <- coord

offset <- 0

for (contig\_ix in 1:nrow(contig\_lengths)) {

on\_contig <- contig == contig\_lengths$contig[contig\_ix]

coord\_flat[on\_contig] <- coord[on\_contig] + offset

offset <- offset + contig\_lengths$length[contig\_ix]

}

coord\_flat

}

Now, we use this to transform the start and end of each window. We also transform the vector of the length of the contigs, so we can use it to add vertical lines between the contigs.

data$start\_flat <- flatten\_coordinates(data$contig,

data$start,

contig\_lengths)

data$end\_flat <- flatten\_coordinates(data$contig,

data$end,

contig\_lengths)

contig\_lengths$length\_flat <- flatten\_coordinates(contig\_lengths$contig,

contig\_lengths$length,

contig\_lengths)

It would be nice to label the x-axis with contig names. One way to do this is to take the coordinates we just made for the vertical lines, add a zero, and shift them one position, like so:

axis\_coord <- c(0, contig\_lengths$length\_flat[-nrow(contig\_lengths)])

Now it’s time to plot! We add one layer of points for the values on the y-axis, where each point is centered on the middle of the window, followed by a layer of vertical lines at the borders between contigs. Finally, we add our custom x-axis, and also some window dressing.

plot\_genome <- ggplot() +

geom\_point(aes(x = (start\_flat + end\_flat)/2,

y = value),

data = data) +

geom\_vline(aes(xintercept = length\_flat),

data = contig\_lengths) +

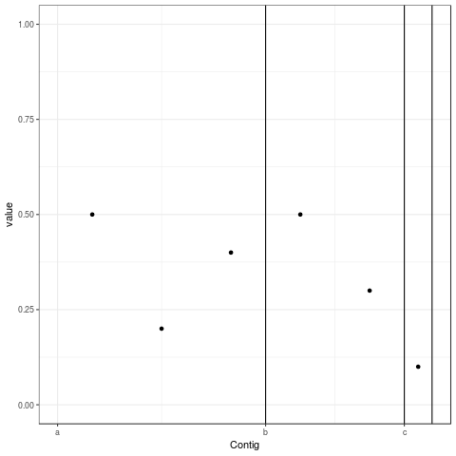
scale\_x\_continuous(breaks = axis\_coord,

labels = contig\_lengths$contig,

limits = c(0, max(contig\_lengths$length\_flat))) +

xlab("Contig") + ylim(0, 1) + theme\_bw()

And this is what we get:

[](https://martinsbioblogg.files.wordpress.com/2019/01/rplot.png)

Its being assured that your plot will look more impressive, but you get the idea.