# Demonstration of Haplotype Visualization

Taylor Anderson

10/26/2020

# Load dependencies

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version
3.6.3
```

```
library(reshape2)
```

```
## Warning: package 'reshape2' was built under R
version 3.6.3
```

# Load core functions

```
# Compare all haplotypes for all sequences to a single
benchmark (pairwise contrasts)
compare_clusters <- function(gt, benchmark, variable){
  wide <- dcast(gt, Chromosome + Positions + dist ~
Sample, value.var=variable)
  out = matrix(nrow = nrow(wide), ncol = ncol(wide))
  colnames(out)<-colnames(wide)
  for(i in 1:nrow(wide)){
    out[i,] <- (wide[i,] == matrix(wide[i,]
[benchmark]))
  }
  out[,1] <- wide[,1]
  out[,2] <- wide[,2]
  out[,3] <- wide[,3]
  out <- melt(data.frame(out),
```

```
id=c("Chromosome","Positions","dist"), check = F)
  colnames(out) <-
c("Chromosome","Positions","dist","Entry","value")
  return(out)
}


# Compare all haplotypes for all sequences to one set
of benchmarks (one-way multi-sample contrasts)
# data is the dataframe, we define a set of "resistant"
samples names along a pedigree
# The function returns a column "Rcluster" (i.e.
matching resistant cluster) with values that are true
(1) if the haplotype clusters match for the resistant
samples
contrast_oneway = function(data, resistant){
  all = c(resistant)
  subset = data[data$Sample %in% all,]
  subset <- dcast(subset, Chromosome + Positions + dist
~ Sample, value.var="hclust")
  #Creates a vector where, if all resistant tomatoes
have the same cluster designation in window, then TRUE
  result = vector(length = nrow(subset))
  for(i in 1:nrow(subset)){
    result[i] <- mean(subset[i,resistant] ==
matrix(subset[i,resistant][1])) == 1
  }
  df <- cbind(subset,"Rcluster" = result)

  for(row in 1:nrow(df)){
  if(df[row,]$Rcluster == FALSE){
    df[row,]$Rcluster <- 0
  } else {
    df[row,]$Rcluster <- 1
    }
```

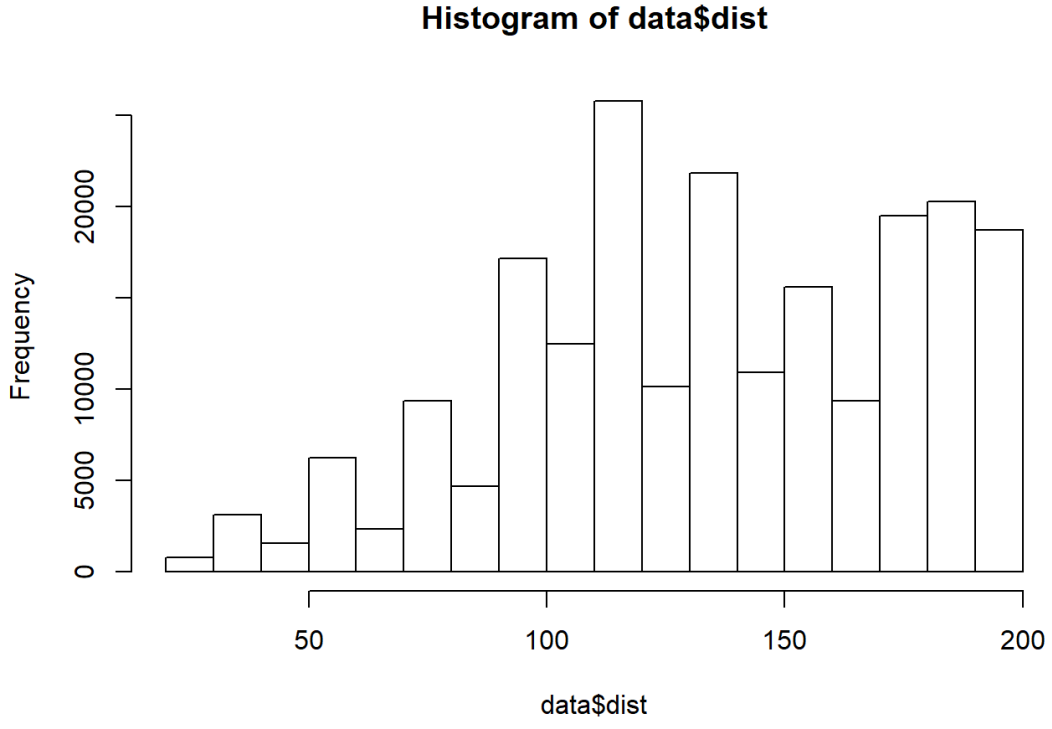
```
  }

  return(df)
}


# Compare all haplotypes for all sequences to two
mutually exclusive sets of benchmarks (two-way multi-
sample contrasts)
# data is the dataframe, we defined the two sets as the
"resistant" and "susceptible" sets, each a list of
multiple sample names
# The function returns a column "Rcluster" (i.e.
matching resistant cluster) with values that are true
(1) if the haplotype clusters for the resistant samples
in that window all match AND none of the susceptible
samples have the same cluster ID as the resistant
samples
contrast_twoway = function(data, resistant,
susceptible){
  all = c(resistant,susceptible)
  subset = data[data$Sample %in% all,]
  subset <- dcast(subset, Chromosome + Positions + dist
~ Sample, value.var="hclust")
  #Creates a vector where, if all resistant tomatoes
have the same cluster designation in window, then TRUE
  res_vec = vector(length = nrow(subset))
  for(i in 1:nrow(subset)){
    res_vec[i] <- mean(subset[i,resistant] ==
matrix(subset[i,resistant][1])) == 1
  }
  #Returns true for all sites where resistant samples
agree and susceptible clusters are not in resistant set
  result = vector(length = length(res_vec))
  for(i in 1:length(res_vec)){
```

```
    if(res_vec[i] == TRUE){
      result[i] = mean(!(subset[i,susceptible] %in%
subset[i,resistant])) == 1
    } else {
      result[i] = FALSE
    }
  }

  df <- cbind(subset,"Rcluster" = result)

  for(row in 1:nrow(df)){
  if(df[row,]$Rcluster == FALSE){
    df[row,]$Rcluster <- 0
  } else {
    df[row,]$Rcluster <- 1
    }
  }

  return(df)
}


#Rename samples from a key-value dataframe (first
column the key and the second the replacement values)
rename_samples <- function(gt, rename){
  gt$Entry <- as.character(gt$Entry)

  for(i in 1:nrow(rename)){
  key <- rename[i,1]
  value <- rename[i,2]
  gt[gt$Entry==key,]$Entry <- value
  }

  gt$Entry <- as.factor(gt$Entry)
  return(gt)
```

```
}
```

# Perform pairwise contrasts relative to a benchmark for a particular chromosomal region

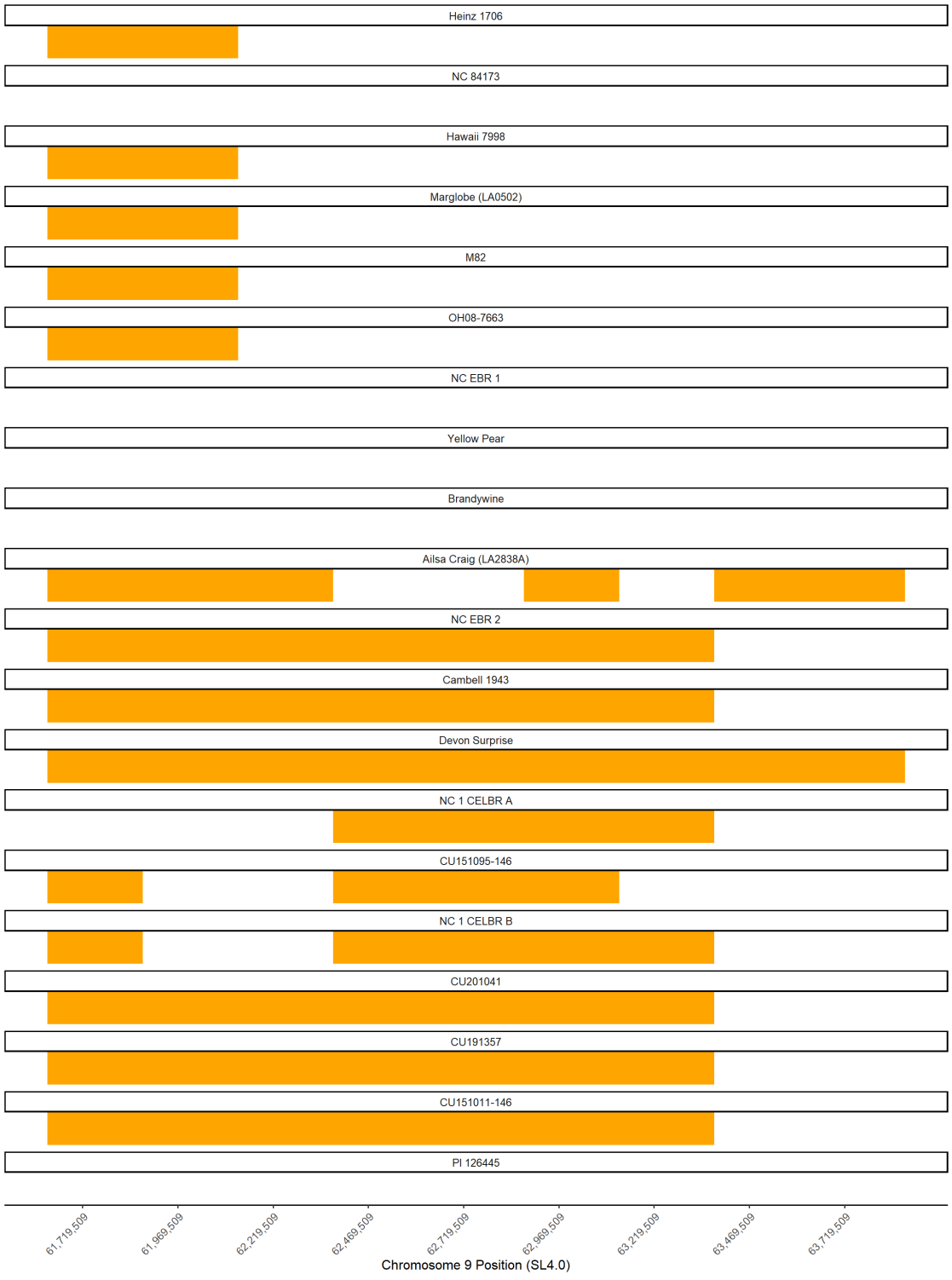
```
#Load the dataset
data=read.csv("./example_files/
ch09_window1000000_step250000_cutoff20_dmin20_dmax200_d
step4_pcacomp2.csv", row.names = 1)

#Look at the distribution of d thresholds
hist(data$dist)
```




```
#Perform the pairwise comparisons relative to sample
```

```
191163 (Devon Surprise)
dataRecode <- compare_clusters(gt=data[data$Chromosome
== "ch09",], benchmark="191163", variable="hclust")

#Load a rename file and rename samples to something
more human readable, then subset by the samples in this
file
rename <- read.table("./example_files/
AccessionRename.txt", sep = "\t", stringsAsFactors = F)
dataRecode <- rename_samples(dataRecode, rename)
subset = subset(dataRecode, Entry %in% rename[,2])
subset$Entry <- factor(subset$Entry, levels =
rename[,2])

#Paint the whole chromosome 9 for a subset of entries,
highlighting prior EB-9 boundaries
windowsize = 1000000
stepSize = 250000

#Optional specification of maximum windows to look at
based on large d thresholds (large values mean less
differentiating genetic information content in window)
- the default value as written below is the maximum
value returned by the algorithm across all windows
dmax = 1*range(data$dist)[2]

#Set the window boundaries to look at
xmin <- 61819509-1e5
xmax <- 63679761+1e5

#Plot the homologous haplotypes by sample relative to
benchmark
ggplot(data=subset[subset$Positions >= xmin &
subset$Positions <= xmax & subset$dist <=dmax,],
aes(x=Positions, y=value)) +
```

```
  theme_classic(base_size = 12) +
  theme(plot.background = element_blank(),
panel.grid.major = element_blank(), panel.grid.minor =
element_blank()) +
  geom_bar(stat="identity", width = stepSize,
fill="orange", colour=NA, size=0) +
  theme(axis.text.x = element_text(angle = 45, hjust =
1), axis.line.y = element_blank(), axis.ticks.y =
element_blank()) +
  ylab(NULL) + xlab("Chromosome 9 Position (SL4.0)") +
  scale_y_continuous(labels = NULL, expand = c(0,0)) +
  facet_wrap(~Entry, ncol = 1) + #Number of columns to
distribute plots across
  scale_x_continuous(labels = scales::comma,
breaks=seq(xmin, xmax, 2.5e5)) #Specify range and size
of x axis labels:
```




```
  #theme(strip.background = element_blank(),
strip.text.x = element_blank()) #option to remove names
(add + above)
```

# Search for and define introgressions in the genome based on a pattern

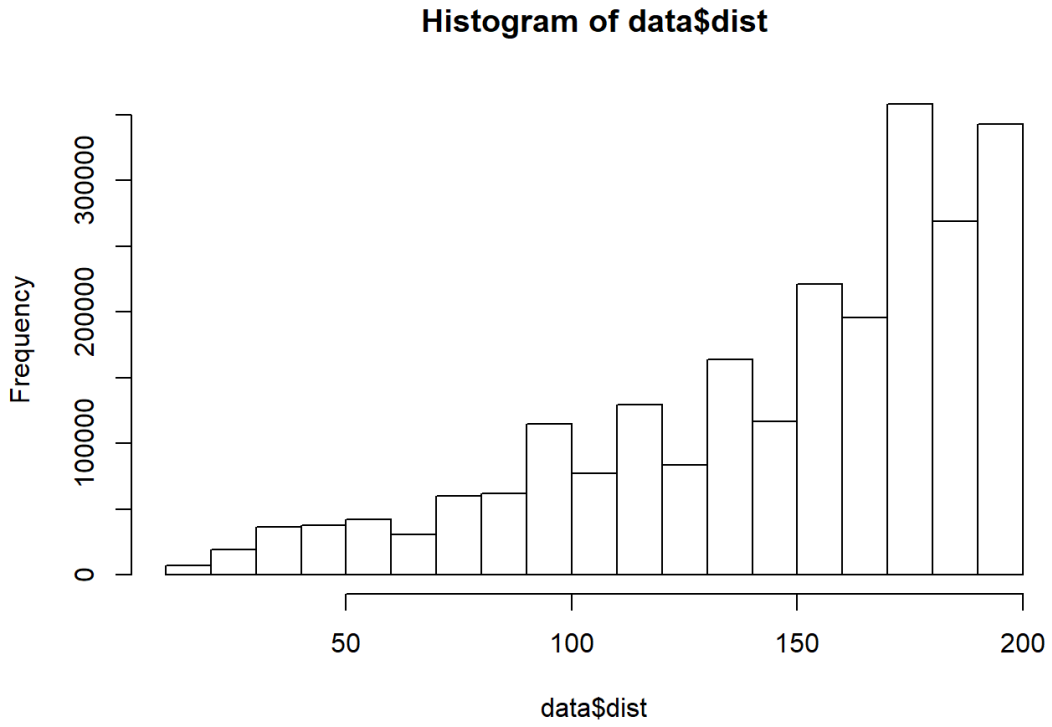
## Prepare a dataset including all chromosome-level files

```
# Merge single chromosome files together (if
applicable)
data01=read.csv("./example_files/
ch01_window1000000_step250000_cutoff20_dmin20_dmax200_d
step4_pcacomp2.csv")
data02=read.csv("./example_files/
ch02_window1000000_step250000_cutoff20_dmin20_dmax200_d
step4_pcacomp2.csv")
data03=read.csv("./example_files/
ch03_window1000000_step250000_cutoff20_dmin20_dmax200_d
step4_pcacomp2.csv")
data04=read.csv("./example_files/
ch04_window1000000_step250000_cutoff20_dmin20_dmax200_d
step4_pcacomp2.csv")
data05=read.csv("./example_files/
ch05_window1000000_step250000_cutoff20_dmin20_dmax200_d
step4_pcacomp2.csv")
data06=read.csv("./example_files/
ch06_window1000000_step250000_cutoff20_dmin20_dmax200_d
step4_pcacomp2.csv")
data07=read.csv("./example_files/
```

```
ch07_window1000000_step250000_cutoff20_dmin20_dmax200_d
step4_pcacomp2.csv")
data08=read.csv("./example_files/
ch08_window1000000_step250000_cutoff20_dmin20_dmax200_d
step4_pcacomp2.csv")
data09=read.csv("./example_files/
ch09_window1000000_step250000_cutoff20_dmin20_dmax200_d
step4_pcacomp2.csv")
data10=read.csv("./example_files/
ch10_window1000000_step250000_cutoff20_dmin20_dmax200_d
step4_pcacomp2.csv")
data11=read.csv("./example_files/
ch11_window1000000_step250000_cutoff20_dmin20_dmax200_d
step4_pcacomp2.csv")
data12=read.csv("./example_files/
ch12_window1000000_step250000_cutoff20_dmin20_dmax200_d
step4_pcacomp2.csv")
all <-
rbind(data01,data02,data03,data04,data05,data06,data07,
data08,data09,data10,data11,data12)
write.csv(all,
"allChr_window1000000_step250000_cutoff20_dmin20_dmax20
0_dstep4_pcacomp2.csv")
```

# Load the concatenated dataset

```
data=read.csv("./example_files/
allChr_window1000000_step250000_cutoff20_dmin20_dmax200
_dstep4_pcacomp2.csv", row.names = 1)
data$hclust <- factor(data$hclust)
hist(data$dist)
```




```
stepSize = 1000000
windowsize = 250000
```

# Search the whole genome for introgressions fitting a one-way pattern

```
#Define the resistant set
resistant =
c('191163','191164','191167','191172','191175')

#Perform a one-way contrast
df <- contrast_oneway(data, resistant)

# The following are the windows fitting the pattern
#df[df$Rcluster==TRUE,]
```

```
#Plot the results
ggplot(data=df) +
  theme_classic(base_size = 12) +
  theme(legend.position = "none", plot.background =
element_blank(),
        panel.grid.major = element_blank(),
panel.grid.minor = element_blank(),
        axis.line.y=element_blank()) +
  geom_tile(aes(x = Positions, y = 0, fill =
as.logical(Rcluster), width = stepSize)) +
  scale_fill_manual(values = c("grey92","orange")) +
  theme(axis.text.x = element_text(angle = 45, hjust =
1)) +
  ylab(NULL) + xlab("Chromosome Position (bp in SL4.0
genome)") +
  scale_y_continuous(breaks=NULL, ) +
  facet_grid( ~ Chromosome, scales = "free", space =
"free_x") +
  scale_x_continuous(breaks=seq(2e7,
range(df$Positions)[2], 2e7)) # Specify range and size
of x axis labels
```

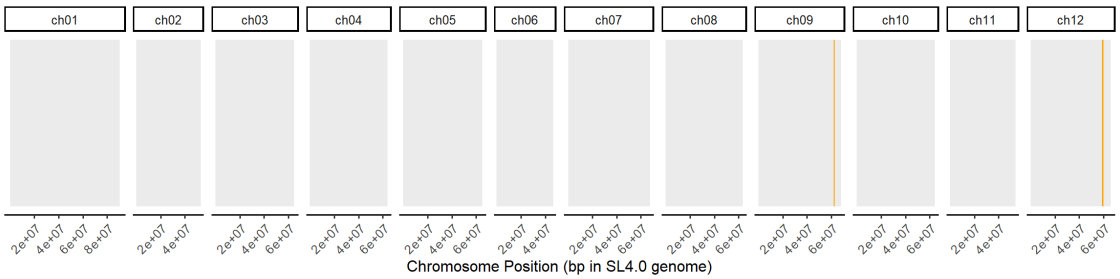

# Search the whole genome for introgressions fitting a two-way pattern

```
#Define the resistant and susceptible sets
```

```
resistant =
c('191163','191164','191167','191172','191175','191174H
','CU3AllData','191357','201041')
susceptible = c('191165','ERR418112','SRR1572598')

#Perform the two way contrast
df <- contrast_twoway(data, resistant, susceptible)

# The following are the windows fitting the pattern
df[df$Rcluster==TRUE,]
##      Chromosome Positions dist 191163 191164 191165
191167 191172 191174H
## 2286       ch09  62502852  116     43     43      1
43     43      43
## 2287       ch09  62752852   96     48     48     23
48     48      48
## 2288       ch09  63002852  188      6      6      4
6      6       6
## 3011       ch12  59501511  124     12     12      0
12     12      12
## 3012       ch12  59751511  124      0      0     17
0      0       0
## 3013       ch12  60001511  148     15     15     19
15     15      15
## 3014       ch12  60251511  116      1      1     27
1      1       1
## 3015       ch12  60501511  104      0      0      6
0      0       0
##      191175 191357 201041 CU3AllData ERR418112
SRR1572598 Rcluster
## 2286     43     43     43         43         1
37        1
## 2287     48     48     48         48        23
23        1
## 2288      6      6      6          6         4
```

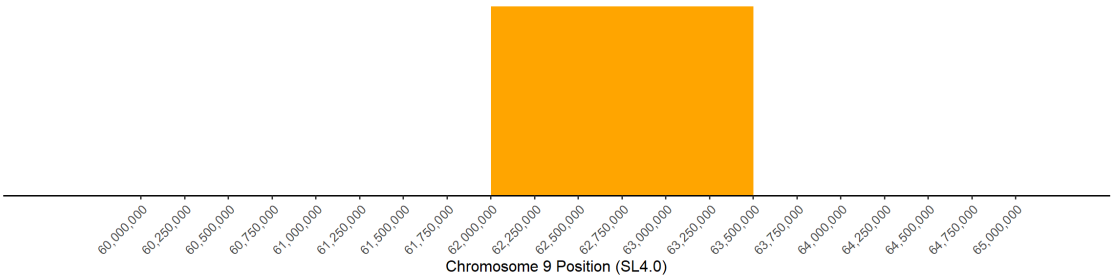
```
4         1
## 3011     12      12      12          12          0
0         1
## 3012      0       0       0           0         17
17        1
## 3013     15      15      15          15         19
19        1
## 3014      1       1       1           1         27
27        1
## 3015      0       0       0           0          6
6         1
```

```
#Plot the homologous haplotypes
ggplot(data=df) +
  theme_classic(base_size = 12) +
  theme(legend.position = "none", plot.background =
element_blank(),
        panel.grid.major = element_blank(),
panel.grid.minor = element_blank(),
        axis.line.y=element_blank()) +
  geom_tile(aes(x = Positions, y = 0, fill =
as.logical(Rcluster), width = stepSize)) +
  scale_fill_manual(values = c("grey92","orange")) +
  theme(axis.text.x = element_text(angle = 45, hjust =
1)) +
  ylab(NULL) + xlab("Chromosome Position (bp in SL4.0
genome)") +
  scale_y_continuous(breaks=NULL, ) +
  facet_grid( ~ Chromosome, scales = "free", space =
"free_x") +
  scale_x_continuous(breaks=seq(2e7,
range(df$Positions)[2], 2e7)) # Specify range and size
of x axis labels
```




# Investigate a zoomed in haplotype region (to better estimate boundaries)

```
#Zoom into one chromosomal region
xmin <- 6e7
xmax <- 6.5e7
chromosome <- "ch09"
ggplot(data=df[df$Chromosome == chromosome &
df$Positions >= xmin & df$Positions <= xmax,],
aes(x=Positions, y=Rcluster)) +
  theme_classic(base_size = 12) +
  theme(plot.background = element_blank(),
panel.grid.minor = element_blank()) +
  geom_bar(stat="identity", width = stepSize,
fill="orange", colour=NA, size=0) +
  theme(axis.text.x = element_text(angle = 45, hjust =
1), axis.line.y = element_blank(), axis.ticks.y =
element_blank()) +
  ylab(NULL) + xlab("Chromosome 9 Position (SL4.0)") +
  scale_y_continuous(labels = NULL, expand = c(0,0)) +
  scale_x_continuous(labels = scales::comma,
breaks=seq(xmin, xmax, 2.5e5))
## Warning: position_stack requires non-overlapping x
intervals
```


60,000,000
60,250,000
60,500,000
60,750,000
61,000,000
61,250,000
61,500,000
61,750,000
62,000,000
62,250,000
62,500,000
62,750,000
63,000,000
63,250,000
63,500,000
63,750,000
64,000,000
64,250,000
64,500,000
64,750,000
65,000,000
Chromosome 9 Position (SL4.0)