# capC-MAP Downstream Analysis in R

*Chris Brackley*

*26 November 2018*

This page shows an example of downstream analysis and plotting using various R packages for Capture-C interaction profiles output from capC-MAP. It uses data obtained from GEO:GSE120666 as an example, with data from two experiments in two different cell types, each with the same three target viewpoints.

The main packages used are :

- ggplot2 : a system for declaratively creating graphics, based on The Grammar of Graphics. Details can be found at https://ggplot2.tidyverse.org/
- Gviz : a bioconductor package for plotting data and annotation information along genomic coordinates, available at http://bioconductor.org/packages/release/bioc/html/Gviz.html
- TxDb.Mmusculus.UCSC.mm9.knownGene and Mus.musculus : bioconductor packages for retrieving gene annotation data available at http://bioconductor.org/packages/TxDb.Mmusculus.UCSC.mm9.knownGene/ and http://bioconductor.org/packages/Mus.musculus/

In addition to this, we will also need the tidyr, gridExtra and data.table packages. To load these in R use the following commands:

```r
library(ggplot2)
library(tidyr)
library(gridExtra)
library(Gviz)
library(TxDb.Mmusculus.UCSC.mm9.knownGene)
library(Mus.musculus)
library(data.table)
```

### Plot per experiment statistics

capC-MAP generates a file 'captured_report.dat' which contains information about read mapping, valid interactions, discarded reads etc. To read this file in R the following commands can be used:

```r
report <- readLines("data1/captured_report.dat")

total <- strtoi(tail(strsplit(report[[8]],split = " ")[[1]],1))
dups <- strtoi(tail(strsplit(report[[12]],split = " ")[[1]],4)[[1]])

inter <- strtoi(tail(strsplit(report[[24]],split = " ")[[1]],4)[[1]])
intra <- strtoi(tail(strsplit(report[[25]],split = " ")[[1]],4)[[1]])

totalvalid <- inter + intra

exclusion <- strtoi(tail(strsplit(report[[20]],split = " ")[[1]],4)[[1]])
multitarget <- strtoi(tail(strsplit(report[[18]],split = " ")[[1]],4)[[1]])
nonmapped <- strtoi(tail(strsplit(report[[16]],split = " ")[[1]],4)[[1]])
noreporter <- strtoi(tail(strsplit(report[[19]],split = " ")[[1]],4)[[1]])
notarget <- strtoi(tail(strsplit(report[[17]],split = " ")[[1]],4)[[1]])
```

Plots showing the level of PCR duplicates, proportions of different types of discarded reads, and the total fraction of intra- and inter- chromosomal reads can be generated with:

```
strip <- theme(legend.position="none") + theme_bw() +
         theme(axis.text=element_blank(),axis.ticks=element_blank(),
               panel.grid=element_blank()) +
         theme(axis.title.x = element_blank(),axis.title.y = element_blank())

df <- data.frame(group=c("Duplicates","Other"),value = c(dups, total))
pie <- ggplot(df, aes(x="", y=value, fill=group)) + geom_bar(width = 1, stat = "identity") +
       coord_polar(theta="y")
p1 <- pie + strip + ggtitle(" \nOf Total Reads")+labs(fill="") +
            guides(fill=guide_legend(nrow=3,byrow=TRUE)) +
            theme(legend.position="bottom",legend.margin = unit(x=c(0,0,0,0),units="mm"))

df <- data.frame(group=c("Mapped","Not Mapped"),
               value = c(totalvalid+exclusion+multitarget, nonmapped+noreporter+notarget))
pie <- ggplot(df, aes(x="", y=value, fill=group)) +
       geom_bar(width = 1, stat = "identity")+coord_polar(theta="y")
p2 <- pie + strip + ggtitle("After Duplicates\nRemoved") + labs(fill="") +
            guides(fill=guide_legend(nrow=3,byrow=TRUE)) +
            theme(legend.position="bottom", legend.margin = unit(x=c(0,0,0,0),units="mm"))

df <- data.frame(group=c("None Mapped","No Report","No Target"),
                 value = c(nonmapped, notarget,noreporter))
pie <- ggplot(df, aes(x="", y=value, fill=group)) + geom_bar(width = 1, stat = "identity") +
       coord_polar(theta="y")
p3 <- pie + strip + ggtitle(" \nOf Unmapped") + labs(fill="") +
            guides(fill=guide_legend(nrow=3,byrow=TRUE)) +
            theme(legend.position="bottom",legend.margin = unit(x=c(0,0,0,0),units="mm"))

df <- data.frame(group=c("Valid Interactions","Multiple Targets","Exclusion Zone"),
                 value = c(totalvalid, multitarget,exclusion))
pie <- ggplot(df, aes(x="", y=value, fill=group)) + geom_bar(width = 1, stat = "identity") +
       coord_polar(theta="y")
p4 <- pie + strip + ggtitle("Of Mapped")+labs(fill="") +
            guides(fill=guide_legend(nrow=3,byrow=TRUE)) +
            theme(legend.position="bottom",legend.margin = unit(x=c(0,0,0,0),units="mm"))

df <- data.frame(group=c("Intrachromosomal","Interchromosomal"),value = c(intra, inter))
pie <- ggplot(df, aes(x="", y=value, fill=group)) + geom_bar(width = 1, stat = "identity") +
       coord_polar(theta="y")
p5 <- pie + strip + ggtitle("Of Valid Interactions")+labs(fill="") +
            guides(fill=guide_legend(nrow=3,byrow=TRUE)) +
            theme(legend.position="bottom",legend.margin = unit(x=c(0,0,0,0),units="mm"))

png("rplots/pie_charts.png")
grid.arrange(p1, p2, p3, p4, p5, nrow=2)
dev.off()
```

### Plot per target statistics

capC-MAP also outputs statistics on each target in the file ‘captured_interactioncounts.dat’, which can be read into R using the following commands:

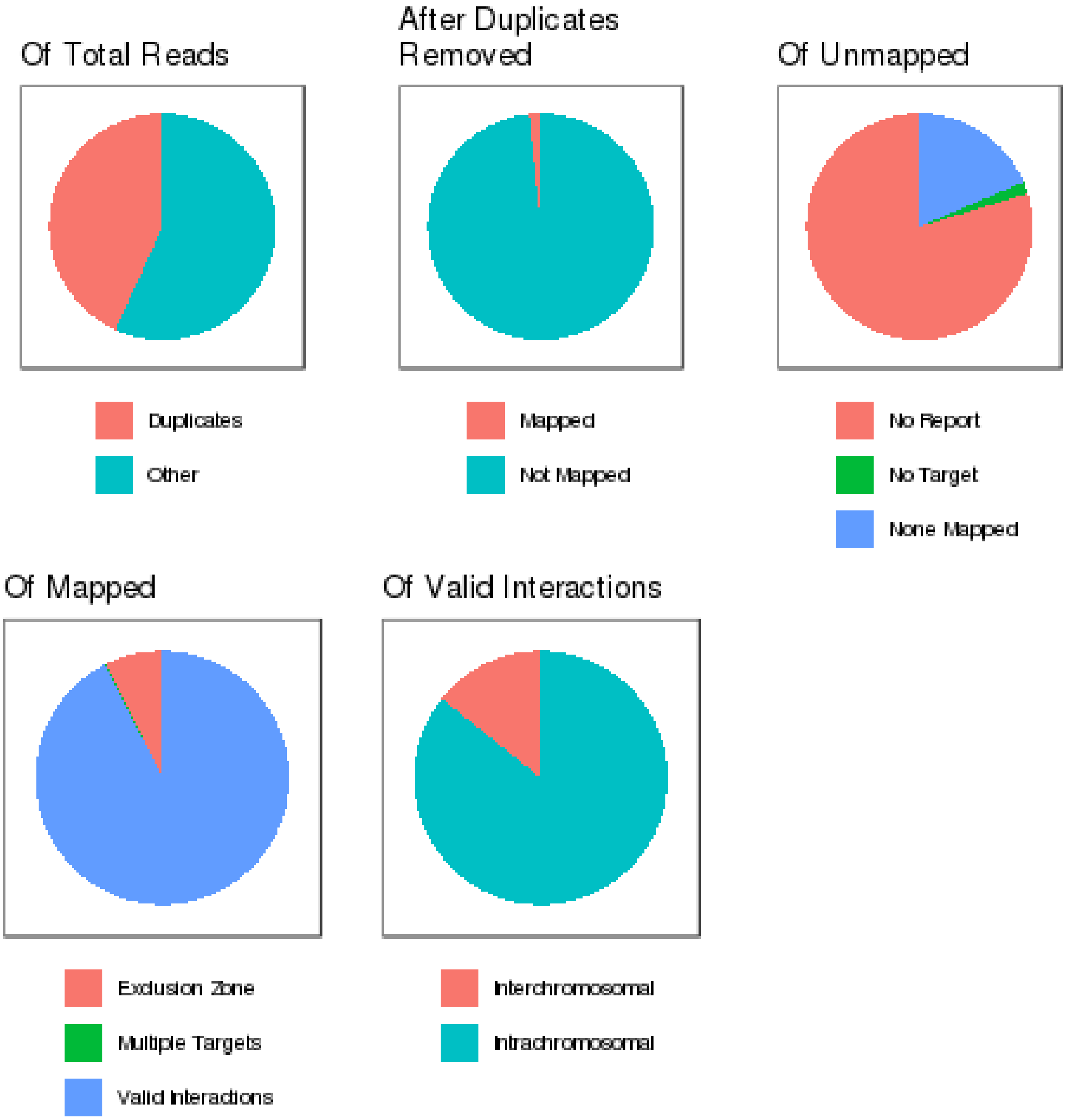

Of Total Reads
Duplicates
Other
After Duplicates Removed
Mapped
Not Mapped
Of Unmapped
No Report
No Target
None Mapped
Of Mapped
Exclusion Zone
Multiple Targets
Valid Interactions
Of Valid Interactions
Interchromosomal
Intrachromosomal

Figure 1:

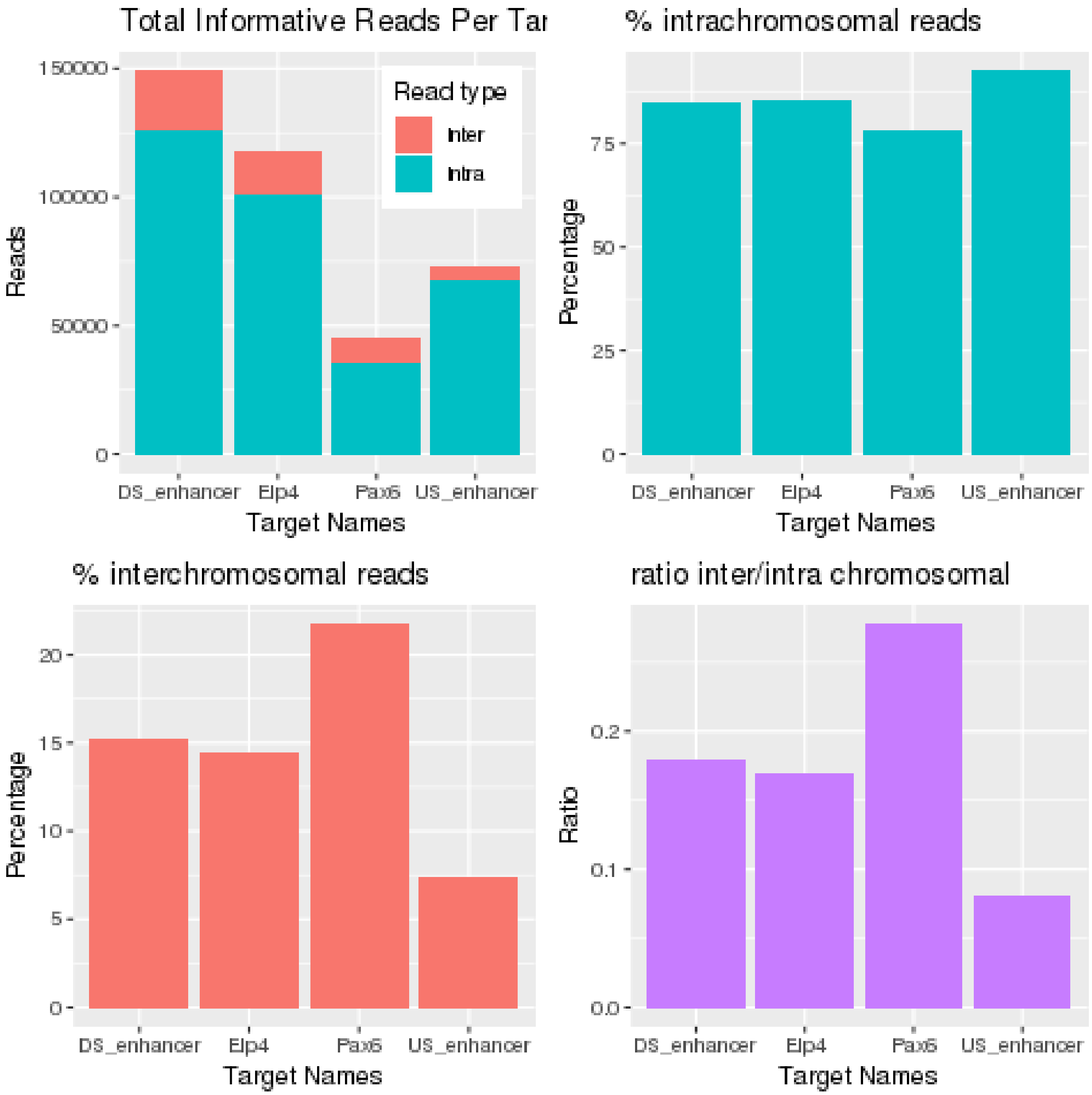
```
pt_stats <- read.table("data1/captured_interactioncounts.dat")
colnames(pt_stats)<-c("target_names","intra_rds","inter_rds","tot_rds","V5","V6")

target_names <- pt_stats$target_names

#calculate percentages and ratios
pt_stats["intra_rds_p"] <- 100*(pt_stats$intra_rds/pt_stats$tot_rds)
pt_stats["inter_rds_p"] <- 100*(pt_stats$inter_rds/pt_stats$tot_rds)
pt_stats["ratio"] <- pt_stats$inter_rds/pt_stats$intra_rds

#convert wide table to long table for stacked bar plot
pt_stats_short<-pt_stats[,1:3]
pt_stats_short <- pt_stats_short %>% gather(inter_rds, intra_rds, -c(target_names))
colnames(pt_stats_short)<-c("target_names","type","rds")
```

To generate bar plots showing interactions:

```
p<-ggplot(data=pt_stats_short, aes(x=target_names, y=rds, fill=type)) + geom_col()
p1 <- p + ggtitle("Total Informative Reads Per Target") + xlab("Target Names") +
          ylab("Reads") +
          scale_fill_discrete(name="Read type", breaks=c("inter_rds", "intra_rds"),
                              labels=c("Inter","Intra"))+
          theme(legend.position = c(0.8, 0.2)) +
          theme(legend.position = c(0.8, 0.8))

p <- ggplot(data=pt_stats, aes(x=target_names, y=intra_rds_p)) +
     geom_bar(stat="identity",fill="#00BFC4")
p2 <- p + ggtitle("% intrachromosomal reads") + xlab("Target Names") + ylab("Percentage")

p <- ggplot(data=pt_stats, aes(x=target_names, y=inter_rds_p)) +
     geom_bar(stat="identity",fill="#F8766D")
p3 <- p + ggtitle("% interchromosomal reads") + xlab("Target Names") + ylab("Percentage")

p <- ggplot(data=pt_stats, aes(x=target_names, y=ratio)) +
     geom_bar(stat="identity",fill="#C77CFF")
p4 <- p + ggtitle("ratio inter/intra chromosomal") + xlab("Target Names") +ylab("Ratio")

png("rplots/bargraphs.png")
grid.arrange(p1, p2, p3, p4, nrow=2)
dev.off()
```

Similarly, plots showing the % of local (<1Mbp) interactions can be generated:

```
pt_stats1 <- read.table("data1/captured_interactioncounts.dat")
pt_stats2 <- read.table("data2/captured_interactioncounts.dat")
colnames(pt_stats1)[1]<-"target_names"
colnames(pt_stats2)[1]<-"target_names"
pt_stats1["local_inter_cell_1"] <- 100*(pt_stats1$V6/pt_stats1$V4)
pt_stats2["local_inter_cell_2"] <- 100*(pt_stats2$V6/pt_stats2$V4)

p <- ggplot(data=pt_stats1, aes(x=target_names, y=local_inter_cell_1)) +
     geom_bar(stat="identity",fill="#00BFC4")
p1 <- p + ggtitle("% local (< 1Mbp interactions)\n cell type 1") + xlab("Target Names") +
          ylab("Percentage")
p <- ggplot(data=pt_stats2, aes(x=target_names, y=local_inter_cell_2)) +
```



Total Informative Reads Per Tar

% intrachromosomal reads

% interchromosomal reads

ratio inter/intra chromosomal

Figure 2:

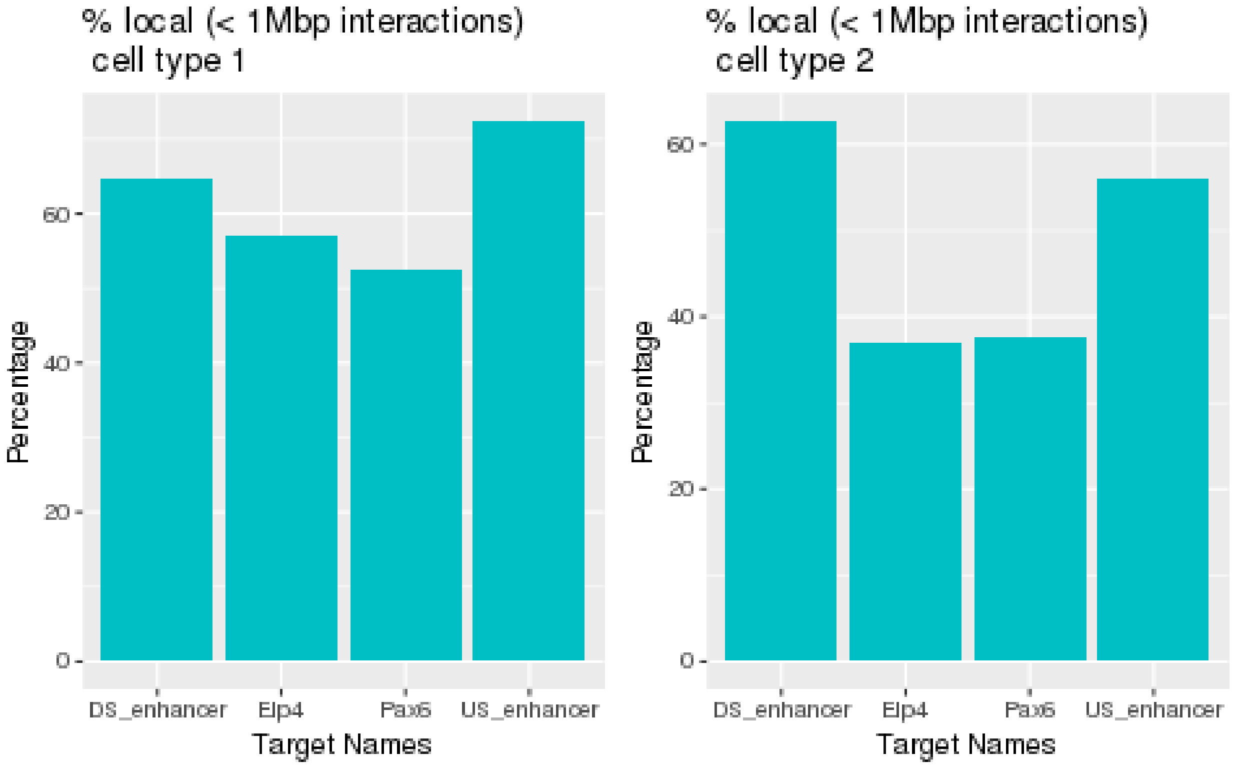


Figure 3:

```
     geom_bar(stat="identity",fill="#00BFC4")
p2 <- p + ggtitle("% local (< 1Mbp interactions)\n cell type 2") + xlab("Target Names") +
          ylab("Percentage")

png("rplots/local_long.png",height=300)
grid.arrange(p1, p2, ncol=2)
dev.off()
```

### Plot interaction profiles with different binning / smoothing parameters

The Gviz package from bioconductor can be used for plotting bed and bedGraph files alongside gene annotations.

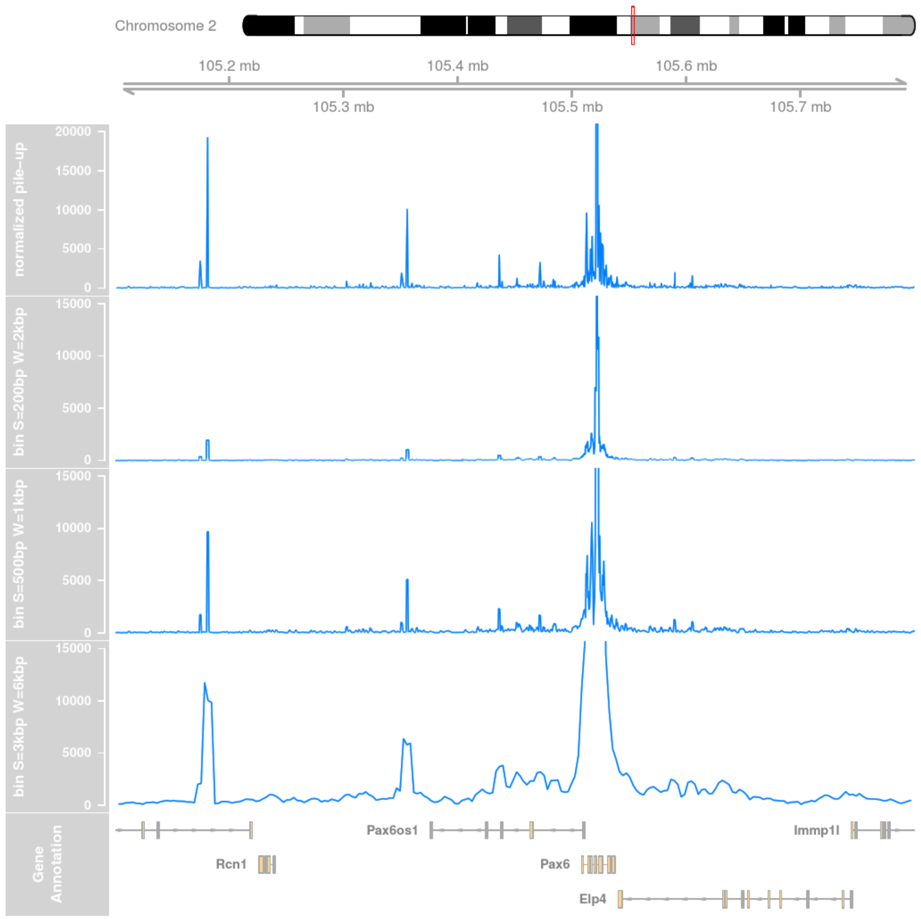
First, set a data range to plot, and load these profiles:

```
thechr <- "chr2"
st <- 105100000
en <- 105800000

txdb<-TxDb.Mmusculus.UCSC.mm9.knownGene

raw_bdg1 <- fread('data1/captured_normalizedpileup_Pax6.bdg',
                  col.names = c('chromosome', 'start', 'end', 'value'))
raw_bdg1_chr <- raw_bdg1[chromosome == thechr]
```

```
raw_bdg2 <- fread(’data1/captured_bin_200_2000_RPM_Pax6.bdg’,
                  col.names = c(’chromosome’, ’start’, ’end’, ’value’))
raw_bdg2_chr <- raw_bdg2[chromosome == thechr]

raw_bdg3 <- fread(’data1/captured_bin_500_1000_RPM_Pax6.bdg’,
                  col.names = c(’chromosome’, ’start’, ’end’, ’value’))
raw_bdg3_chr <- raw_bdg3[chromosome == thechr]

raw_bdg4 <- fread(’data1/captured_bin_3000_6000_RPM_Pax6.bdg’,
                  col.names = c(’chromosome’, ’start’, ’end’, ’value’))
raw_bdg4_chr <- raw_bdg4[chromosome == thechr]
```

To generate plots of the interaction profile from one of the targets at different binning and smoothing levels:

```
dtrack1 <- DataTrack(range = raw_bdg1_chr, type = "a",genome = ’mm9’,
                     name = "normalized pile-up", ylim = c(0,20000))
dtrack2 <- DataTrack(range = raw_bdg2_chr, type = "a",genome = ’mm9’,
                     name = "bin S=200bp W=2kbp", ylim = c(0,15000))
dtrack3 <- DataTrack(range = raw_bdg3_chr, type = "a",genome = ’mm9’,
                     name = "bin S=500bp W=1kbp", ylim = c(0,15000))
dtrack4 <- DataTrack(range = raw_bdg4_chr, type = "a",genome = ’mm9’,
                     name = "bin S=3kbp W=6kbp", ylim = c(0,15000))
itrack <- IdeogramTrack(genome = "mm9", chromosome = thechr)
gtrack <- GenomeAxisTrack()

grtrack <- GeneRegionTrack(txdb,chromosome = thechr, start = st, end = en,
                           name = "Gene Annotation",collapseTranscripts = ’meta’,
                           geneSymbols=TRUE,stackHeight=0.5,min.height=1)
# Add gene names instead of id numbers
symbols <- unlist(mapIds(org.Mm.eg.db, gene(grtrack),
                         "SYMBOL", "ENTREZID", multiVals = "first"))
symbol(grtrack) <- symbols[gene(grtrack)]

pdf("rplots/bin_interactions.pdf")
plotTracks(list(itrack, gtrack, dtrack1,dtrack2,dtrack3,dtrack4, grtrack),
          from = st, to = en)
dev.off()
```

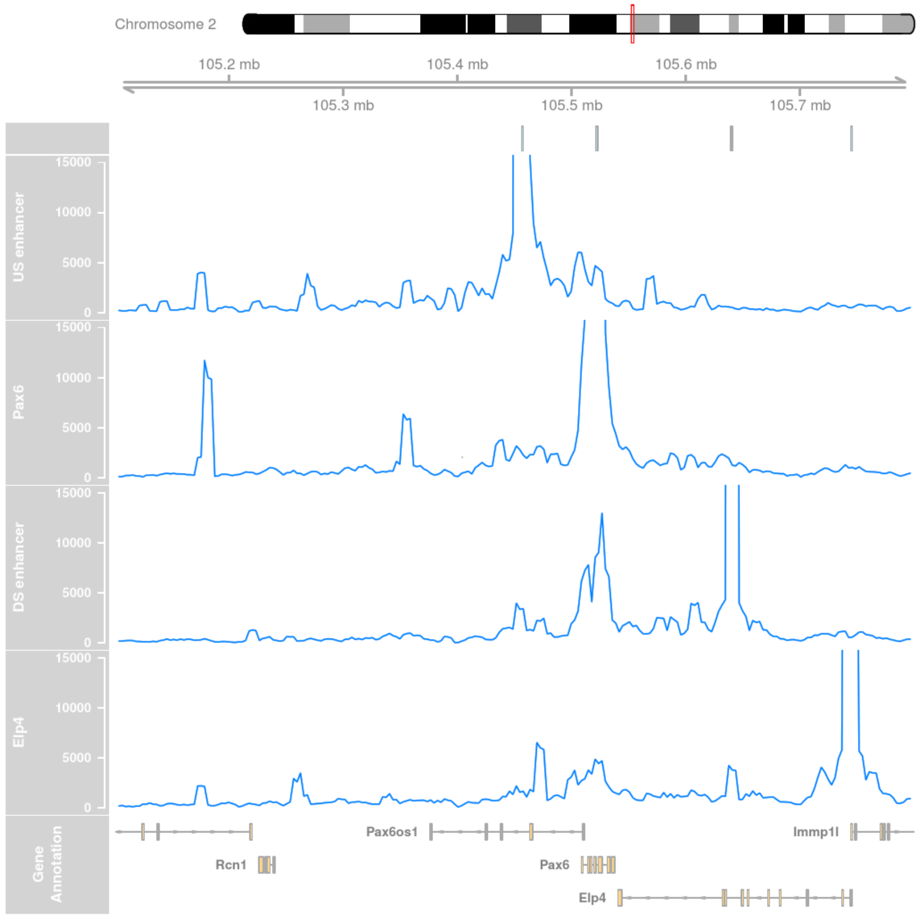


### Plot interaction profiles from different targets

Instead plot the interaction profiles for several targets:

```r
thechr <- "chr2"
st <- 105100000
en <- 105800000

raw_bdg1 <- fread('data1/captured_bin_3000_6000_RPM_US_enhancer.bdg',
                  col.names = c('chromosome', 'start', 'end', 'value'))
raw_bdg1_chr <- raw_bdg1[chromosome == thechr]

raw_bdg2 <- fread('data1/captured_bin_3000_6000_RPM_Pax6.bdg',
                  col.names = c('chromosome', 'start', 'end', 'value'))
raw_bdg2_chr <- raw_bdg2[chromosome == thechr]

raw_bdg3 <- fread('data1/captured_bin_3000_6000_RPM_DS_enhancer.bdg',
                  col.names = c('chromosome', 'start', 'end', 'value'))
raw_bdg3_chr <- raw_bdg3[chromosome == thechr]

raw_bdg4 <- fread('data1/captured_bin_3000_6000_RPM_Elp4.bdg',
                  col.names = c('chromosome', 'start', 'end', 'value'))
raw_bdg4_chr <- raw_bdg4[chromosome == thechr]
```

```
dtrack1 <- DataTrack(range = raw_bdg1_chr, type = "a",genome = ’mm9’,name = "US enhancer",
                     ylim = c(0,15000))
dtrack2 <- DataTrack(range = raw_bdg2_chr, type = "a",genome = ’mm9’,name = "Pax6",
                     ylim = c(0,15000))
dtrack3 <- DataTrack(range = raw_bdg3_chr, type = "a",genome = ’mm9’,name = "DS enhancer",
                     ylim = c(0,15000))
dtrack4 <- DataTrack(range = raw_bdg4_chr, type = "a",genome = ’mm9’,name = "Elp4",
                     ylim = c(0,15000))
itrack <- IdeogramTrack(genome = "mm9", chromosome = thechr)
gtrack <- GenomeAxisTrack()

targets <- fread(’data1/targets.bed’,col.names=c(’chromosome’,’start’,’end’,’V4’,’name’))
ttrack <- AnnotationTrack(targets,shape=’box’,chromosome=thechr,name="targets",
                          stack.height=0.25)

pdf("rplots/target_interactions.pdf")
plotTracks(list(itrack, gtrack,ttrack, dtrack1,dtrack2,dtrack3,dtrack4, grtrack),
           from = st, to = en)
dev.off()
```


Chromosome 2
105.2 mb
105.4 mb
105.6 mb
105.3 mb
105.5 mb
105.7 mb
US enhancer
Pax6
DS enhancer
Elp4
Gene Annotation
Pax6os1
Immp1l
Rcn1
Pax6
Elp4