CLK refactor and test of robustness

Introduction

CLK-dependent exon recognition and conjoined gene formation revealed with a novel small molecule inhibitor

CLK encodes a member of the CDC2-like (or LAMMER) family of dual specificity protein kinases. In the cell nucleus, the encoded protein phosphorylates serine/arginine-rich proteins involved in pre-mRNA processing, releasing them into the nucleoplasm. The choice of splice sites during pre-mRNA processing may be regulated by the concentration of transacting factors, including serine/arginine-rich proteins. Therefore, the encoded protein may play an indirect role in governing splice site selection.

The authors describe a new compound, T3, a CLK small molecule inhibitor which shows superior potentcy and selectivity than the current standard, KH-CB19.

Of particular interest in the formation of conjoined genes (CG) when CLK is inhibited.

Sequencing

```
library(knitr)
library(rmarkdown)
library(ggplot2)
library(stringr)
library(biomaRt)
library(Gviz)
```

```
## Loading required package: S4Vectors
```

```
## Loading required package: stats4
```

```
## Loading required package: BiocGenerics
```

```
## Loading required package: parallel
```

```
##
## Attaching package: 'BiocGenerics'
```

```
## The following objects are masked from 'package:parallel':
##
##     clusterApply, clusterApplyLB, clusterCall, clusterEvalQ,
##     clusterExport, clusterMap, parApply, parCapply, parLapply,
##     parLapplyLB, parRapply, parSapply, parSapplyLB
```

```
## The following objects are masked from 'package:stats':
##
##     IQR, mad, sd, var, xtabs
```

```
## The following objects are masked from 'package:base':
##
##     anyDuplicated, append, as.data.frame, basename, cbind, colnames,
##     dirname, do.call, duplicated, eval, evalq, Filter, Find, get, grep,
##     grepl, intersect, is.unsorted, lapply, Map, mapply, match, mget,
```

```
##     order, paste, pmax, pmax.int, pmin, pmin.int, Position, rank,
##     rbind, Reduce, rownames, sapply, setdiff, sort, table, tapply,
##     union, unique, unsplit, which.max, which.min

##
## Attaching package: 'S4Vectors'

## The following object is masked from 'package:base':
##
##     expand.grid

## Loading required package: IRanges

## Loading required package: GenomicRanges

## Loading required package: GenomeInfoDb

## Loading required package: grid
```

```
library(org.Hs.eg.db)
```

```
## Loading required package: AnnotationDbi

## Loading required package: Biobase

## Welcome to Bioconductor
##
##     Vignettes contain introductory material; view with
##     'browseVignettes()'. To cite Bioconductor, see
##     'citation("Biobase")', and for packages 'citation("pkgname")'.

##
```

```
library(TxDb.Hsapiens.UCSC.hg38.knownGene)
```

```
## Loading required package: GenomicFeatures
```

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following object is masked from 'package:AnnotationDbi':
##
##     select

## The following object is masked from 'package:Biobase':
##
##     combine

## The following objects are masked from 'package:GenomicRanges':
##
##     intersect, setdiff, union

## The following object is masked from 'package:GenomeInfoDb':
##
##     intersect

## The following objects are masked from 'package:IRanges':
##
##     collapse, desc, intersect, setdiff, slice, union

## The following objects are masked from 'package:S4Vectors':
##
```

```
##     first, intersect, rename, setdiff, setequal, union

## The following objects are masked from 'package:BiocGenerics':
##
##     combine, intersect, setdiff, union

## The following object is masked from 'package:biomaRt':
##
##     select

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
metadata<-read.csv("metadata/metadata.csv")

# for my notes: Oo1Zafe3Qox3
```

`nrow(metadata)` were obtained

Breakdowns

```
metadata %>% group_by(Platform) %>% summarize(samples=n(),median_reads=median(spots))
```

By platform

```
## # A tibble: 2 x 3
##   Platform    samples median_reads
##   <chr>         <int>        <dbl>
## 1 ILLUMINA        109       795864
## 2 PACBIO_SMRT      60        81741
```

By cell line HCT116 is a human colorectal carcinoma (i.e. malignant or "transformed") cell line. 184-hTERT-L2 cell line is derived from human mammary epithelial cells immortalized by transduction with hTERT.

```
metadata$cell_line<-as.vector(str_extract_all(metadata$SampleName,'(HCT116|184-hTert)',simplify=TRUE))
metadata %>% group_by(cell_line) %>% summarize(samples=n()) %>% arrange(-samples)
```

```
## # A tibble: 2 x 2
##   cell_line samples
##   <chr>       <int>
## 1 HCT116        161
## 2 184-hTert       8
```

By treatment Various silencing transfections were tested on CLK itself, splicing factors such as U2AF2 and SRSF9, TIA1 and CPEB RNA-binding proteins siCLK, siCPEB ,siDAZA, siHNRNPH, siKHDRBS1, siLIN28A, siELAVL1, siHNRNPC, siHNRNPF, siU2AF2, siTIA, siSRSF, siSRRM, siSFPQ, siSAMD4B, RNA recognition motif (RRMs) and splicing enhancers are also tested.

```
metadata %>% group_by(SampleName) %>% summarize(samples=n()) %>% arrange(-samples)
```

```
## # A tibble: 90 x 2
##    SampleName                  samples
##    <chr>                         <int>
##  1 0.5 uM T3 treated HCT116         24
##  2 5 uM T3 treated HCT116           24
##  3 Untreated HCT116                 24
##  4 1.0 uM T3 treated HCT116          4
##  5 0.05 uM T3 treated HCT116         2
##  6 0.1 uM T3 treated HCT116          2
##  7 0.5 uM T3 treated 184-hTert       2
##  8 1.0 uM T3 treated 184-hTert       2
##  9 10 uM T3 treated HCT116           2
## 10 5.0 uM T3 treated 184-hTert       2
## # ... with 80 more rows
```

Alignment and rMATS-ISO

Alignment was performed with STAR against hg38 using GTEX pipeline settings. The alignments themselves were run on the Truwl.com platform.

```
STAR --runMode alignReads \
     --outSAMtype BAM SortedByCoordinate \
     --limitBAMsortRAM ${bytes} \
     --readFilesCommand zcat \
     --outFilterType BySJout   --outFilterMultimapNmax 20 \
     --outFilterMismatchNmax 999   --alignIntronMin 25  \
     --alignIntronMax 1000000   --alignMatesGapMax 1000000 \
     --alignSJoverhangMin 8   --alignSJDBoverhangMin 5 \
     --sjdbGTFfile GRCh38_star/genes.gtf \
     --genomeDir GRCh38_star \
     --runThreadN ${cpus} \
     --outFileNamePrefix ${sample}.  \
     --readFilesIn  ${sample}_1.fastq.gz ${sample}_2.fastq.gz
```

RMATS-ISO was run using the gencode.v28.annotation.gtf, roughly equivalent to the Ensembl gtf.

RMATS-EM output

The following columns are returned from rMATS-EM:

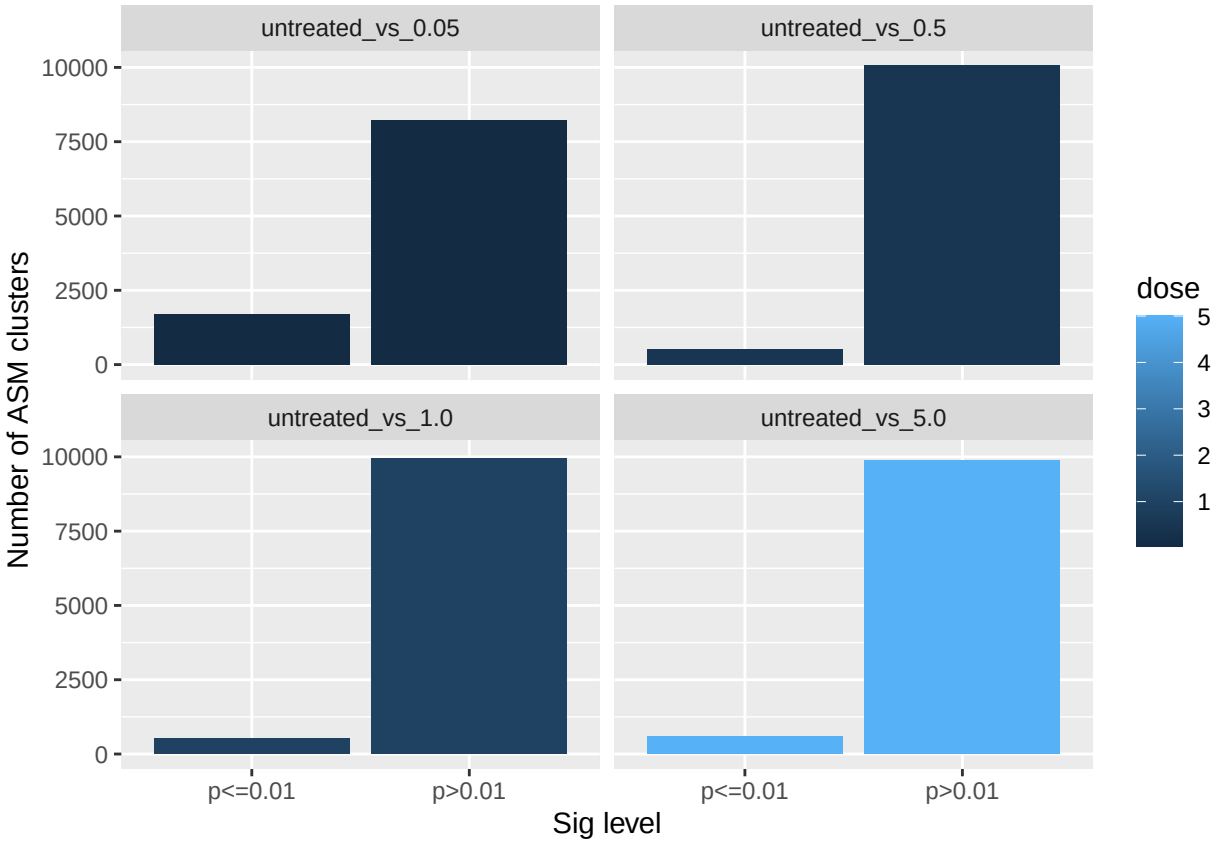
```
 [1] "ASM_name"                       "total_isoforms"                 "total_exons"
 [6] "p_value"                        "test_statistic"                 "isoform_inclusion_group_1"
[11] "variance_group_1"               "variance_group_2"               "variance_constrained"
[16] "dirichlet_parameter_constrained" "paired_isoform_pvalues"         "isoform_index"
```

Dose dependent splicing patterns of T3 in HCT116

```
pthresh<-0.01
doses<-c('0.05','0.5','1.0','5.0')
control<-"untreated"
```

```
em_all<-NULL
for(dose in doses){
```

```
  em<-read.table(paste0("results/iso_",control,'_vs_',dose,'/EM_out/EM.out'),comment.char = '',strip.wh
  em$dose<-as.numeric(dose)
  em$dosestr<-paste0(control,'_vs_',dose)
  if(!is.null(em_all)){
    em_all<-rbind(em_all,em)
  }else{
    em_all<-em
  }
}

#set p_value of 0 to a token dummy minimum, display only highly significant assemblies
#ggplot(em_all %>% filter(!is.na(p_value)) %>% filter(p_value<=0.01) %>% rowwise() %>% dplyr::mutate(p_
ggplot(em_all %>% dplyr::filter(!is.na(p_value)) %>% rowwise() %>% mutate(sig=ifelse(p_value<=pthresh,pa
```




```
em_all %>% dplyr::filter(!is.na(p_value)) %>% rowwise() %>% mutate(sig=p_value<=pthresh) %>% group_by(d
```

```
## `summarise()` has grouped output by 'dose'. You can override using the `.groups` argument.
```

```
## Using cnt as value column: use value.var to override.
```

```
dimnames(sig_table)[[2]]<-c(paste0("p>",pthresh),paste0("p<=",pthresh))
sig_table<-cbind(sig_table,total=rowSums(sig_table))
sig_table<-cbind(sig_table,sig_frac=round(sig_table[,2]/sig_table[,3],2))
fold_change<-c(0,sapply(1:(nrow(sig_table)-1),function(x){(sig_table[x+1,2]/sig_table[x+1,3])/(sig_table
sig_table<-cbind(sig_table,fold_change=fold_change)
knitr::kable(sig_table)
```

	p>0.01	p<=0.01	total	sig_frac	fold_change
0.05	8228	1698	9926	0.17	0.0000000
0.5	10063	516	10579	0.05	0.2851292
1	9953	516	10469	0.05	1.0105072
5	9888	585	10473	0.06	1.1332879

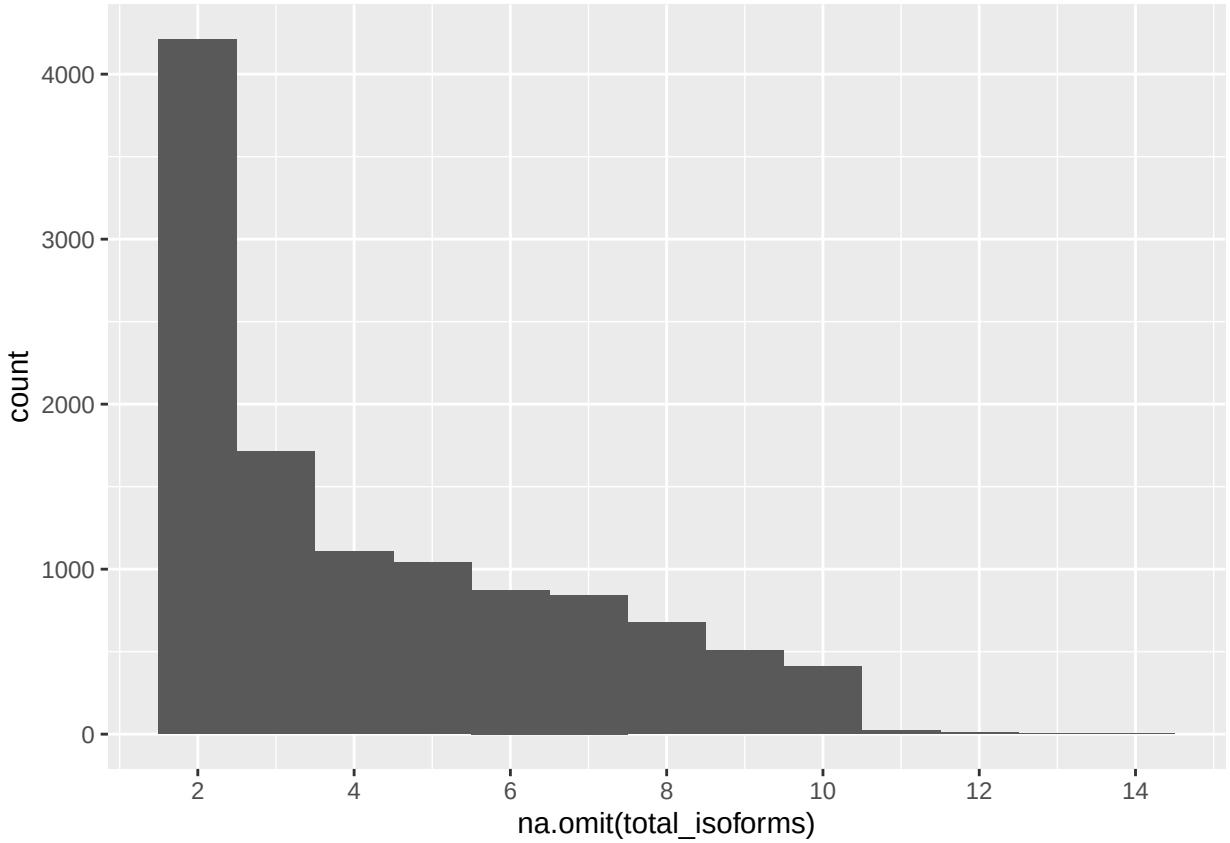
The number of assemblies in which significant AS events (p<0.01) were observed as a fraction of all assemblies was highest in the `untreated_vs_1.0` group, although the greatest fold increase occurs at 0.5uM 1.1332879 reported in the paper as 4.1.

Clusters of interest in T3 0.5

```
dose<-c('0.5')
control<-"untreated"
em<-read.table(paste0("results/iso_",control,'_vs_',dose,'/EM_out/EM.out'),comment.char = '',strip.whit
coor<-read.table(paste0("results/iso_",control,'_vs_',dose,'/ISO_classify/ISO_module_coor.txt'),comment
gene<-read.table(paste0("results/iso_",control,'_vs_',dose,'/ISO_classify/ISO_module_gene.txt'),comment
type<-read.table(paste0("results/iso_",control,'_vs_',dose,'/ISO_classify/ISO_module_type.txt'),comment
#typesummary<-
```

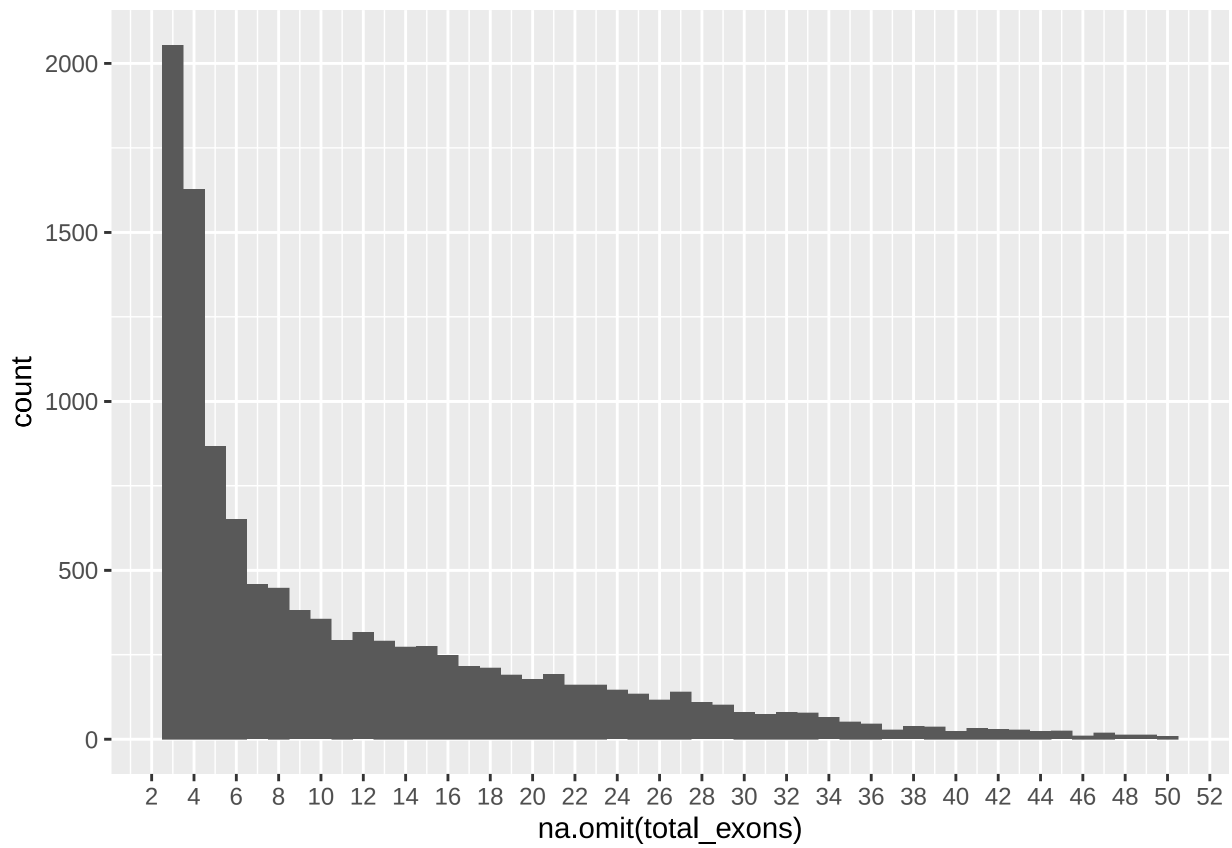
Number of isoforms in T3 0.5 clusters

```
ggplot(em,aes(na.omit(total_isoforms)))+geom_histogram(binwidth=1)+scale_x_continuous(breaks = scales::
```



Number of exons in T3 0.5 clusters

```
ggplot(em,aes(na.omit(total_exons)))+geom_histogram(binwidth=1)+scale_x_continuous(breaks = scales::pret
```



Top 10 simple events

Look at only those clusters with 3 or fewer isoforms, 4 or fewer exons and rank by the lowest paired isoform pvalue among them.

```
strmin<-function(x){
  if(is.na(x)){return(NA)}
  if(x=='NA,NA'){return(NA)}
  return(min(as.numeric((str_split(x,',',simplify = TRUE)))))
}

em %>% dplyr::filter(total_isoforms<=3,total_exons<=4) %>% arrange(p_value) %>% head(n=10) -> simple_eve
```

Top 10 by test statistic

```
knitr::kable(em %>% dplyr::filter(test_statistic==max(em$test_statistic,na.rm = TRUE)))
```

ASM [illegible]
ASM#[illegible] 04,0.00[illegible] 04,0.0[illegible] 04,0,0.0039,0,0.0071,0,0,0,0,0,0

```
em %>% dplyr::filter(test_statistic==max(em$test_statistic,na.rm = TRUE)) %>% pull(ASM_name) %>% as.cha
str_replace(top_hit,'#','') -> top_hit_nohash
gene %>% dplyr::filter(asm==top_hit_nohash)
```

```
##       asm  hugo            ensg
## 1 ASM6705 GAPDH ENSG00000111640
```

```
coords<-coor[which(em$ASM_name==top_hit),]
```

Find conjoined genes

```
library(TxDb.Hsapiens.UCSC.hg38.knownGene)
txdb<-TxDb.Hsapiens.UCSC.hg38.knownGene
genes_gr <- genes(txdb)
```

```
##   1613 genes were dropped because they have exons located on both strands
##   of the same reference sequence or on more than one reference sequence,
##   so cannot be represented by a single genomic range.
##   Use 'single.strand.genes.only=FALSE' to get all the genes in a
##   GRangesList object, or use suppressMessages() to suppress this message.
```

```
#don't want overlapping genes
reduce(genes_gr) -> genes_reduced
ir<-IRanges(start=coor$start,end = coor$end)
coor_gr<-GRanges(seqnames = coor$chr,ranges=ir,strand = coor$strand)
GenomicRanges::findOverlaps(subject=genes_reduced,query=coor_gr) -> hits

#this is my apporac to find clusters that span more than one gene
as.data.frame(hits) %>% group_by(queryHits) %>% summarize(nhit=n()) %>% dplyr::filter(nhit>1) %>% pull(

#now with the subset of ranges that span more than one gene hit up teh original txdb so we can get real
cg_gr<-coor_gr[cg,]
GenomicRanges::findOverlaps(subject=genes_gr,query=cg_gr) -> cg_hits

cgranges<-genes_gr[subjectHits(cg_hits),"gene_id"]
```

There are only 25 such conjoined genes identified with clusters that span more than two genes.

#get teh gene ids

```
txtable = biomaRt::select(txdb, keys=cgranges$gene_id, columns=columns(txdb), keytype="GENEID")
```

```
## 'select()' returned 1:many mapping between keys and columns
```

```
library(org.Hs.eg.db)
hgnc_names<-biomaRt::select(org.Hs.eg.db, cgranges$gene_id, "SYMBOL")
```

```
## 'select()' returned 1:1 mapping between keys and columns
```

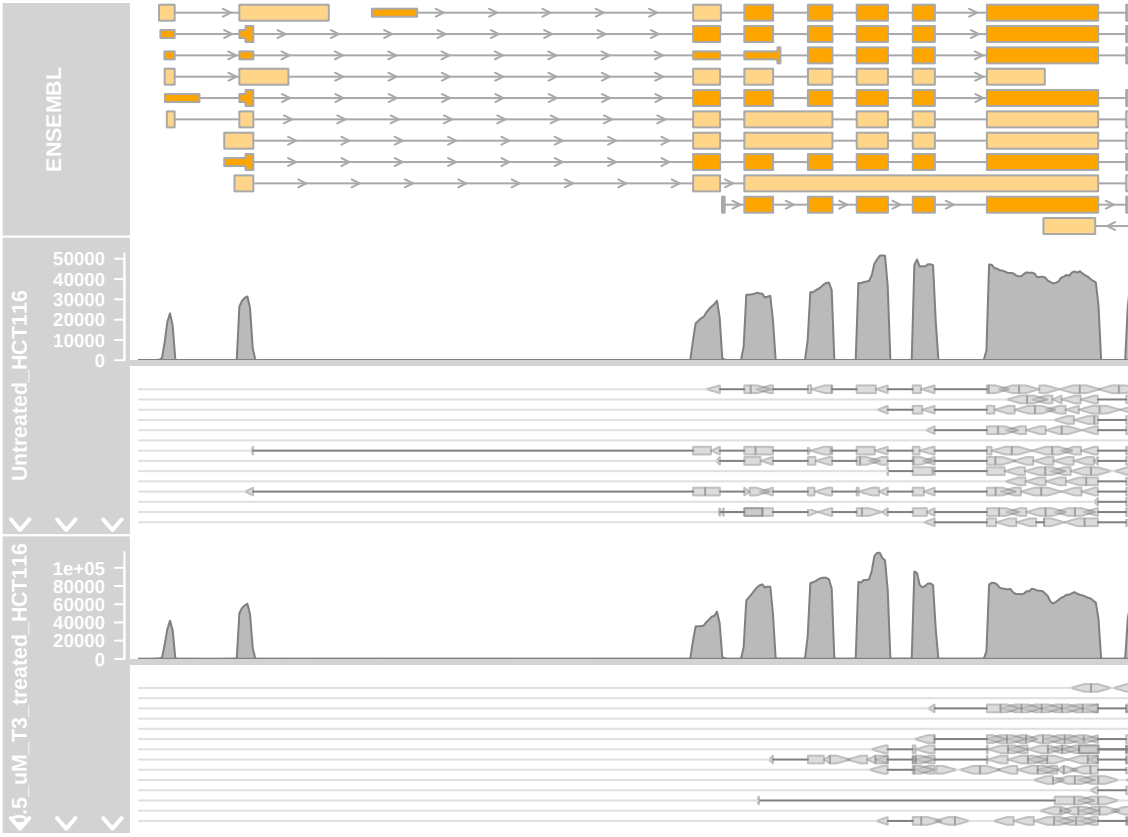
```
cgranges$hgnc<-hgnc_names$SYMBOL
knitr::kable(cgranges)
```

seqnames	start	end	width	strand	gene_id	hgnc
chr3	9933822	9945413	11592	+	78987	CRELD1
chr3	9917074	9933630	16557	+	84818	IL17RC
chr6	41789896	41895361	105466	-	25862	USP49

seqnames	start	end	width	strand	gene_id	hgnc
chr6	41905354	41921139	15786	-	9477	MED20
chr6	85607785	85643792	36008	-	10492	SYNCRIP
chr6	85676637	85678748	2112	-	26799	SNORD50A
chr6	85650491	85678932	28442	-	387066	SNHG5
chr6	85505496	85615234	109739	-	57231	SNX14
chr6	85677589	85677658	70	-	692088	SNORD50B
chr7	27106184	27140225	34042	-	3200	HOXA3
chr7	27128507	27130780	2274	-	3201	HOXA4
chr7	27141052	27143681	2630	-	3202	HOXA5
chr7	27145396	27150603	5208	-	3203	HOXA6
chr7	141764097	141765197	1101	+	50831	TAS2R3
chr7	141778442	141780819	2378	+	50832	TAS2R4
chr8	1801125	1801192	68	+	100500912	MIR3674
chr8	1817231	1817307	77	+	693181	MIR596
chr12	49002274	49018807	16534	-	5571	PRKAG1
chr12	49018975	49059774	40800	-	8085	KMT2D
chr13	27255064	27255135	72	+	26771	SNORD102
chr13	27255401	27255526	126	+	619499	SNORA27
chr15	50354959	50356034	1076	+	100129387	GABPB1-AS1
chr15	50360329	50360410	82	+	100616396	MIR4712
chrX	45746157	45746266	110	-	407006	MIR221
chrX	45747015	45747124	110	-	407007	MIR222

```
dose="0.5"
#treated
metadata %>% dplyr::filter(str_detect(SampleName,dose)) %>% dplyr::filter(str_detect(SampleName,'T3')) %
metadata %>% dplyr::filter(str_detect(SampleName,'Untreated')) %>% dplyr::filter(Platform=='ILLUMINA') %

#aws s3 cp s3://clk-splicing/SRP091981/SRR5009487.Aligned.sortedByCoord.out.bam SRP091981/
atreated<-"SRR5009487"
auntreated<-"SRR5009474"

plot_window<-function(asm,type){
  stringr::str_replace(asm,'#','') -> asm_nohash
  gene %>% dplyr::filter(asm==asm_nohash)
  coords<-coor[which(em$ASM_name==asm),]
  afrom <- coords$start - 100
  ato <- coords$end + 100
  chr <- as.character(coords$chr)
  treatedName <- metadata %>% dplyr::filter(Run==atreated) %>% pull(SampleName) %>% as.character() %>% s
  untreatedName <- metadata %>% dplyr::filter(Run==auntreated) %>% pull(SampleName) %>% as.character() %
  if(type=='gviz'){
    treatedTrack <- AlignmentsTrack(paste0("./SRP091981/",atreated,".Aligned.sortedByCoord.out.md.bam")
    untreatedTrack  <- AlignmentsTrack(paste0("./SRP091981/",auntreated,".Aligned.sortedByCoord.out.md.
    options(ucscChromosomeNames=TRUE)

    bmt <- BiomartGeneRegionTrack(genome = "hg38", name="ENSEMBL", chromosome = chr, start = afrom, end
    plotTracks(list(bmt,untreatedTrack,treatedTrack), from = afrom, to = ato, chromosome = chr) #, type
  }else{
    sashimiPlot<-paste0("rmats2sashimiplot --b1 ",paste0("./SRP091981/",atreated,".Aligned.sortedByCoor
```

```
    cat(sashimiPlot)
  }
}

plot_window(top_hit,"gviz")
```



Top hit: ASM#6705