Coding Tasks Gviz - 18th February 2021

Mario Keller

February 19, 2021

Contents

1 Introduction

In today's coding task we want to have a look on the Gviz package. The package offers a flexible framework for the visualization of high-throughput data (e.g. RNA-seq or iCLIP) in the context of different annotations (e.g. Gencode annotation or annotated binding sites). One advantage compared to visualization via IGV is that Gviz enables user-specific adjustment of the visualization and generation of (nearly) publication-ready figures.

The concept used by Gviz is similar to that of a GenomeBrowser, where individual types of genomic features or genomic data are represented by separate tracks. In Gviz these are typically GeneRegionTracks, AlignmentTracks, AnnotationTracks or DataTracks. Each of these track types has a Gviz function creating them (e.g.GeneRegionTrack() or AnnotationTrack()). The returned objects can afterwards be combined in a list and used as input in the plotTracks() function, which visualizes the generated tracks.

A Gviz user guide is available here.

2 Required libraries

The following libraries are required in today's coding tasks.

```
library(Gviz)
library(GenomicRanges)
library(rtracklayer)
library(GenomicFeatures)
```

To install these libraries use the following command.

```
if (!requireNamespace("BiocManager", quietly = TRUE))
    install.packages("BiocManager")

BiocManager::install("Gviz")
```

3 Required data

I uploaded a ZIP-File in my Hessenbox that is accessible here.

The ZIP-File contains

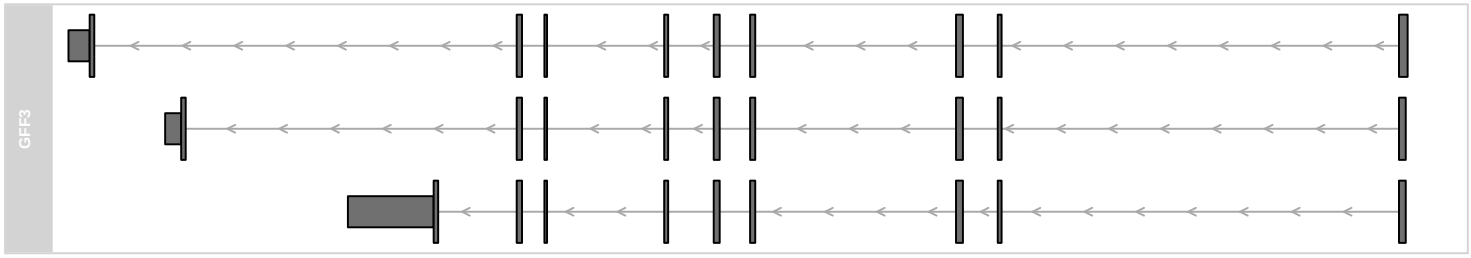
- A GFF3-File containing the annotation of the LAMP2 gene (LAMP2_anno.gff3).
- A BAM-File containing RNA-seq reads from a HNRNPH1 knockdown experiment (KD.H1.bam) and the corresponding BAI-File.
- A BAM-File containing RNA-seq reads from a control experiment (Ctrl.bam) and the corresponding BAI-File.
- A BigWig-File containing the iCLIP binding signal of HNRNPH1 (HNRNPH1_iCLIP.bw).
- A BED-File containing binding sites compute from the HNRNPH1 iCLIP data (HNRNPH1_BSs.bed).

4 Task1 - GeneRegionTrack

In the first step we want to visualize the annotation of the LAMP2 gene. For this purpose we first import the GFF3-File using makeTxDbFromGFF() from the GenomicFeatures package, which creates a TxDB-obj. This object then be used as input for the GeneRegionTrack() function. The returned GeneRegionTrack object can afterwards be visualized with the plotTracks() function. You should also define the region you want to plot using the *from*, *to* and *chromosome* parameter in the plotTracks() function. The region we want to have a look at is gene locus of LAMP2 ("chrX:120427600-120471000").

```
LAMP2.anno.txdb <- makeTxDbFromGFF("data/LAMP2_anno.gff3")
LAMP2.anno.track <- GeneRegionTrack(LAMP2.anno.txdb, name = "GFF3", col = "black", fill = "#707070",
                                    rotation.title=90, cex.title=0.5, frame=T)

plotTracks(LAMP2.anno.track, from=120427600, to=120471000, chromosome="chrX")
```

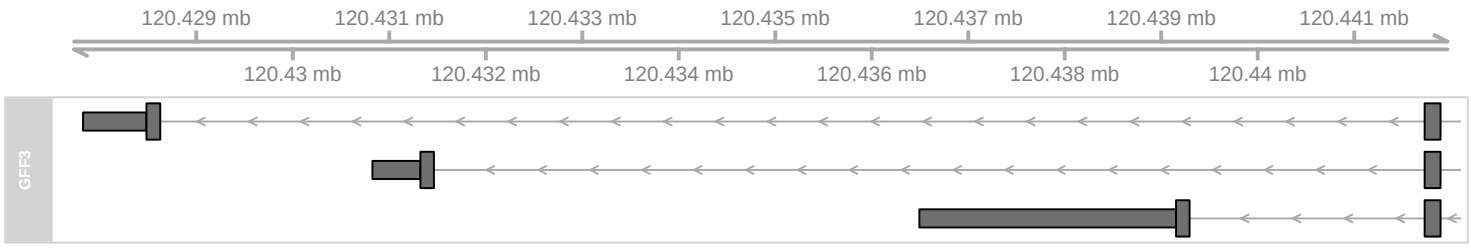


As you can see there are three different isoforms annotated from LAMP2. These three isoforms only differ in their last exon, which covers the 3'end of the CDS and the 3'UTR. From a splicing perspective we define this as an alternative last exon (ALE). To obtain a better resolution, we only want to focus on the region covering the three ALEs and the upstream exon ("chrX:120427600-120442100")

```
plotTracks(LAMP2.anno.track, from=120427600, to=120442100, chromosome="chrX")
```

It is also helpful to see the genomic coordinates. The GenomeAxisTrack() function creates a track illustrating the genomic coordinates and doesn't require any input. Just add it to the list that is used as input for the plotTracks() function.

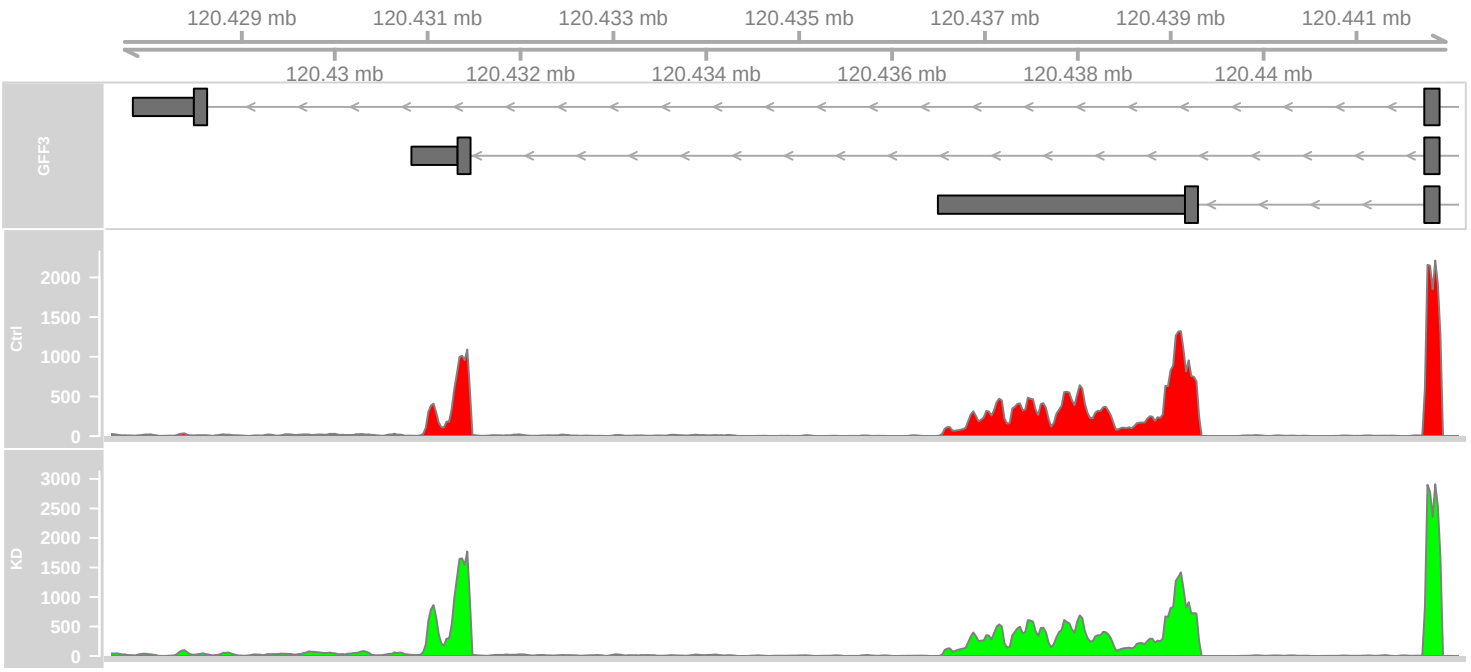
```
plotTracks(list(GenomeAxisTrack(), LAMP2.anno.track), from=120427600, to=120442100, chromosome="chrX")
```



5 Task2 - AlignmentTracks

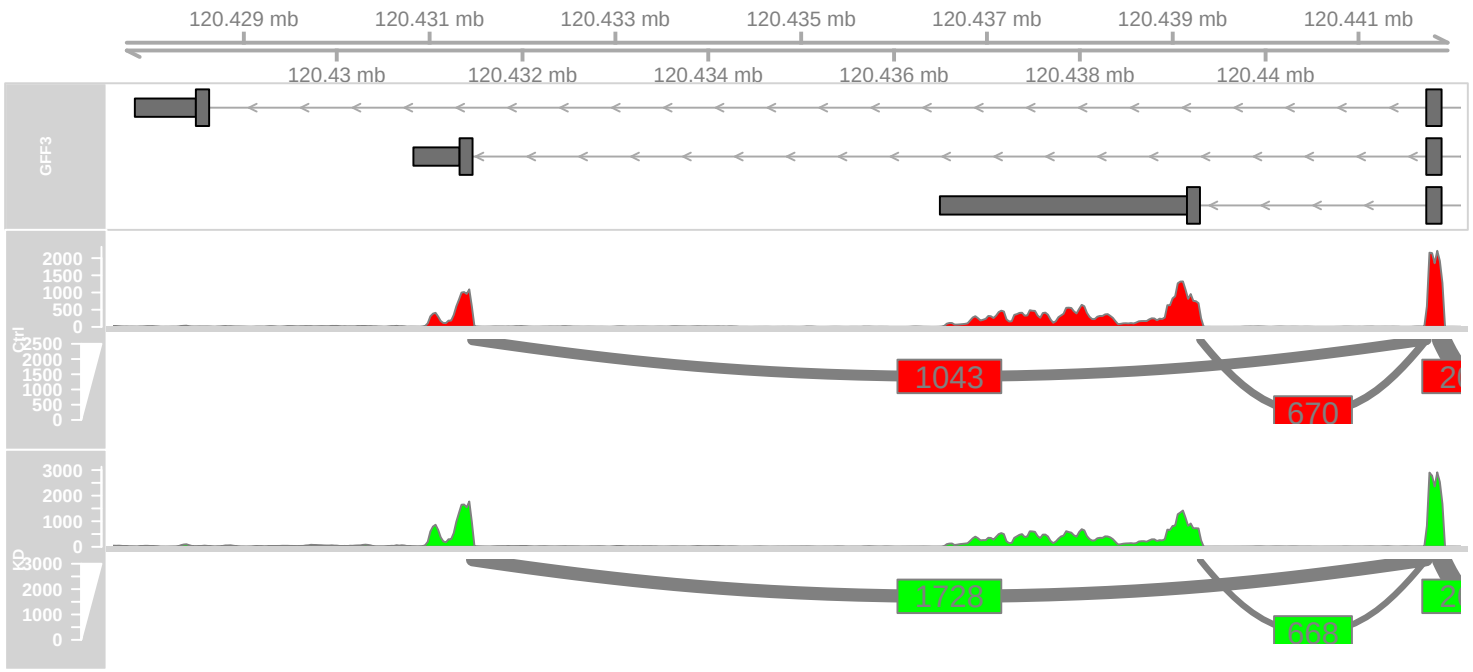
Next, we want to have a look on the RNA-seq data that compromises a HNRNPH1 KD and the corresponding Control. Create for the Ctrl and KD AlignmentTracks() using the provided BAM-Files and add them to the list used as input in plotTracks(). Remove the aligned reads and only visualize the coverage by using *coverageOnly=T* and *type="coverage"* in the AlignmentTracks() function. Control the height of the tracks by defining *sizes=c(1,2,3,3)* in the plotTracks() function.

```
Ctrl.track <- AlignmentsTrack("data/Ctrl.bam", name = "Ctrl", fill = "red", cex.title=0.5, coverageOnly=T,
                              type="coverage")
KD.track <- AlignmentsTrack("data/KD.H1.bam", name = "KD", fill = "green", cex.title=0.5, coverageOnly=T,
                            type="coverage")
plotTracks(list(GenomeAxisTrack(), LAMP2.anno.track, Ctrl.track, KD.track), from=120427600, to=120442100,
           chromosome="chrX", sizes=c(1,2,3,3))
```



It seems that the coverage on the first and second ALE (from right to left) is altered between the Ctrl and the KD. Set *type=c("coverage", "sashimi")* and *sashimiNumber=T* in the AlignmentsTrack() calls. This enables us to see all splice-junctions present in the RNA-seq data and the number of reads supporting them. As there may be a lot of junctions, you can additionally use *sashimiScore=25* to remove all junctions with less than 25 supporting reads.

```
Ctrl.track <- AlignmentsTrack("data/Ctrl.bam", name = "Ctrl", fill = "red", cex.title=0.5, coverageOnly=T,
                              type=c("coverage", "sashimi"), sashimiScore = 25, sashimiNumbers=T)
KD.track <- AlignmentsTrack("data/KD.H1.bam", name = "KD", fill = "green", cex.title=0.5, coverageOnly=T,
                            type=c("coverage", "sashimi"),
                            sashimiScore = 25, sashimiNumbers=T)
plotTracks(list(GenomeAxisTrack(), LAMP2.anno.track, Ctrl.track, KD.track), from=120427600, to=120442100,
           chromosome="chrX", sizes=c(1,2,3,3))
```



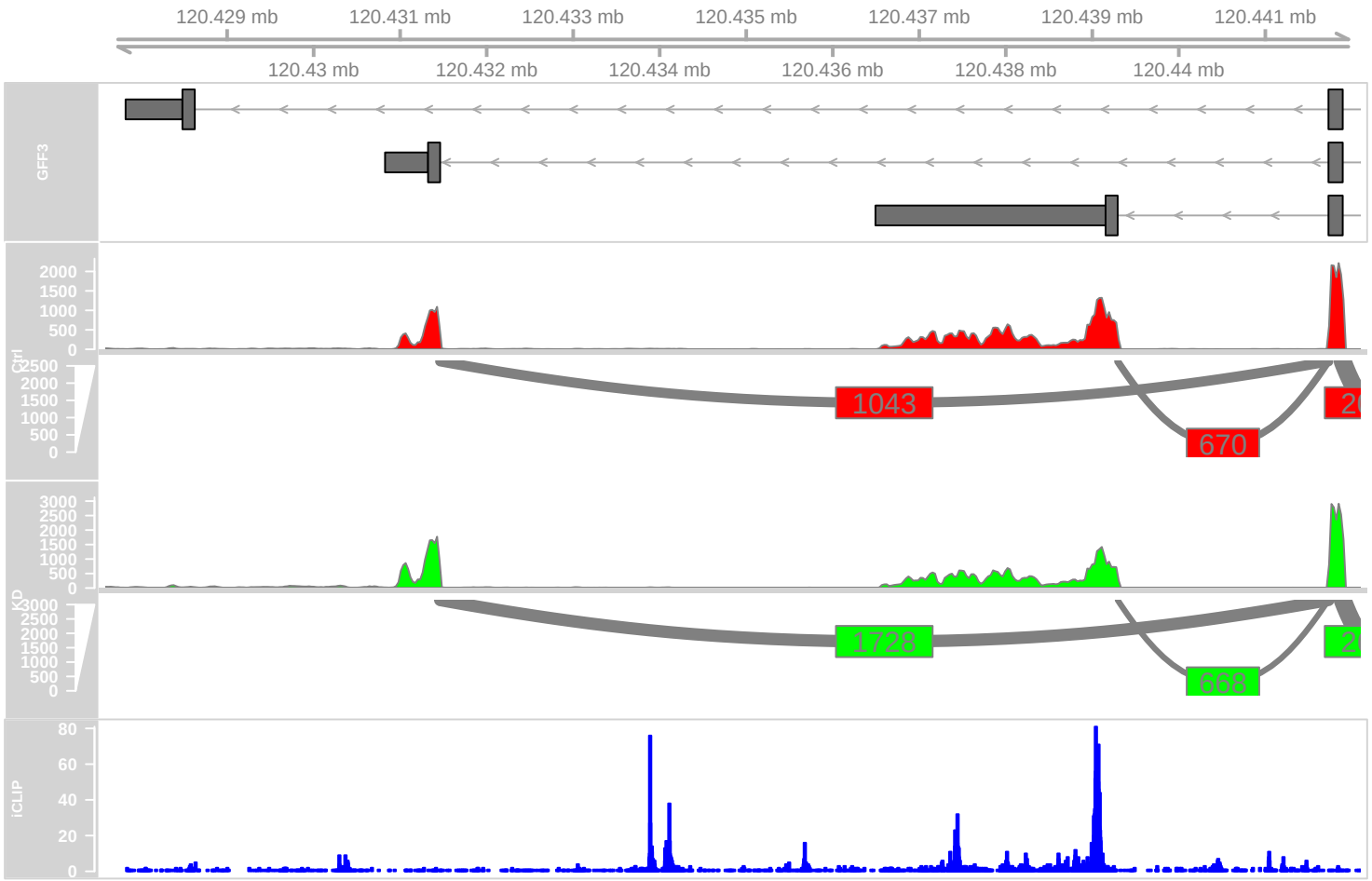
When we compare the KD against the Control, we can clearly see that the KD leads to more skipping of the first ALE and more inclusion of the second ALE.

6 Task3 - DataTrack

Yet we do not know whether the observed change in the usage of the first and second ALE is due to a direct dependency on HNRNPH1 or whether it is an indirect effect. To answer this question we will additionally include results from an iCLIP experiment, which we will visualize as a DataTrack using the DataTrack() function. In addition to the BigWig-File path use *type="histogram"* as input.

```
iCLIP.track =  DataTrack("data/HNRNPH1_iCLIP.bw", name = "iCLIP", fill = "blue", col.histogram = "blue",
                         type="histogram", rotation.title=90, cex.title=0.5, , frame=TRUE)

plotTracks(list(GenomeAxisTrack(), LAMP2.anno.track, Ctrl.track, KD.track, iCLIP.track), from=120427600,
           to=120442100, chromosome="chrX", sizes=c(1,2,3,3,2))
```



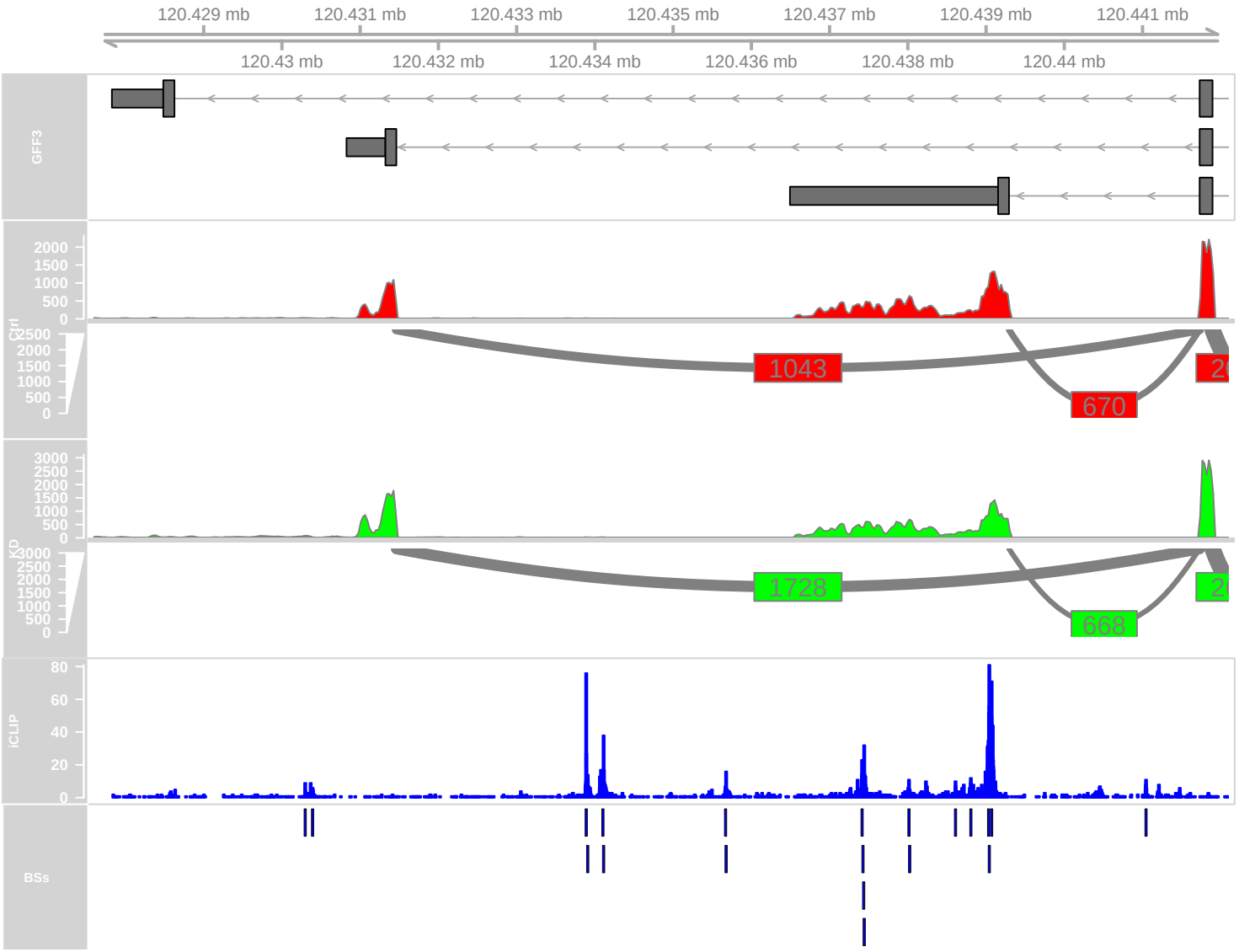
Now we can see that HNRNPH1 has a strong binding signal in the first ALE and in the downstream intron.

7 Task 4 - AnnotationTrack

Although we can clearly see that there is strong binding we do not know where the actual binding sites are located. This information you typically get from so-called peak callers, such as PureCLIP. The binding sites reported by PureClip are provided in the BED-File. This file can be used as input for the import() function from rtracklayer package to create a GRanges object, which can afterwards be used as input for the AnnotationTrack() function. Use *shape="box"* in the AnnotationTrack() function.

```
BS.gr <- import("data/HNRNPH1_BSs.bed")
BS.track =  AnnotationTrack(BS.gr, name = "BSs", shape = "box", fill = "blue", col = "black", lwd = 0.5,
                            rotation.title=0, cex.title=0.5)

plotTracks(list(GenomeAxisTrack(), LAMP2.anno.track, Ctrl.track, KD.track, iCLIP.track, BS.track),
           from=120427600, to=120442100, chromosome="chrX", sizes=c(1,2,3,3,2,2))
```



Now we can eas ily see the binding sites of HNRNPH1, which could be further analyzed to decipher the mechanism affecting the inclusion of the ALEs.