Visualizing genomic features with the Gviz package

Florian Hahne
Novartis Institute for Biomedical Research
Preclinical Safety Informatics

NOVARTIS

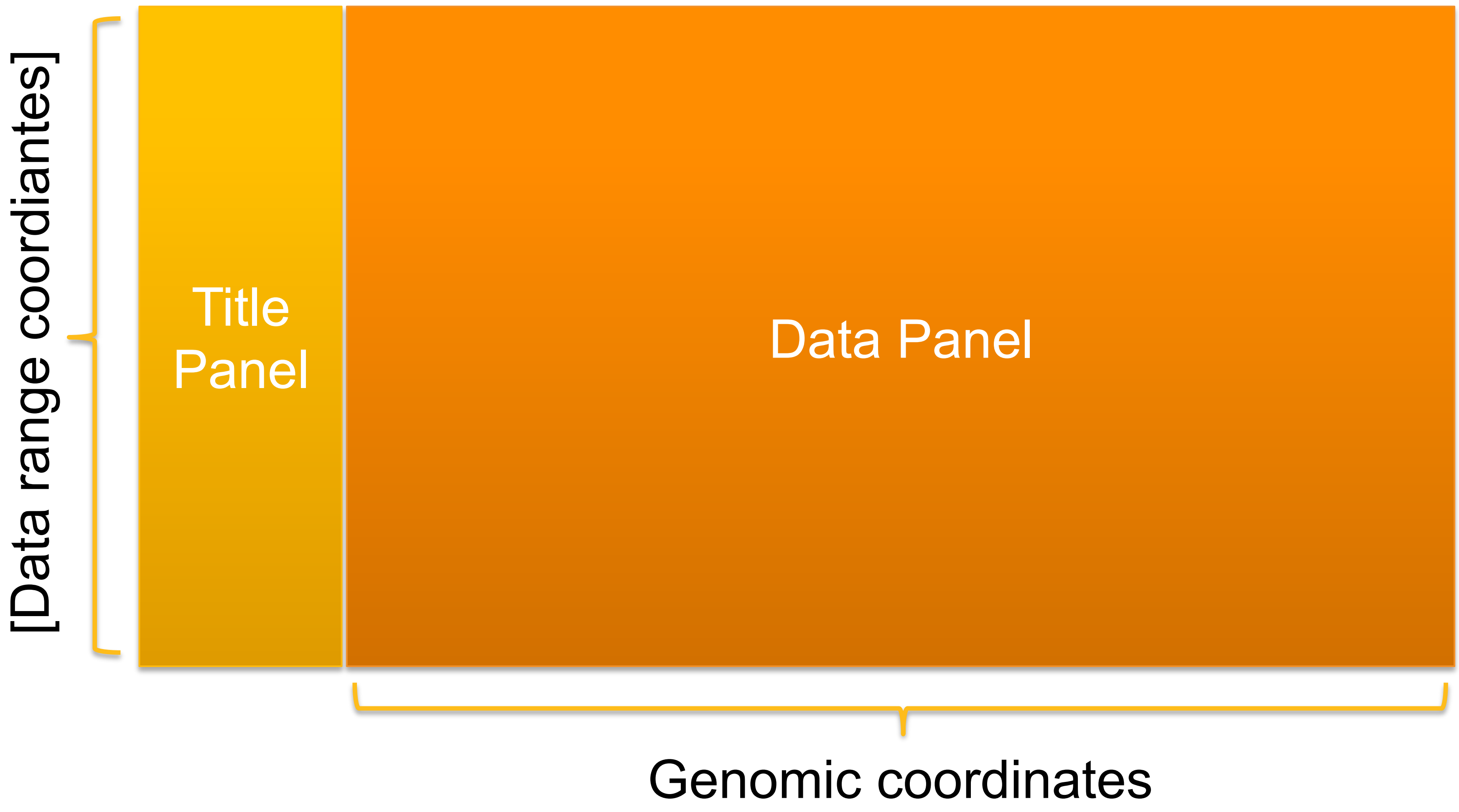
Gviz package: objectives

- High quality visualization of potentially large numeric data along genomic coordinates
  - different views, deal with multiple samples and sample groups
- Additional annotation features (transcripts, SNPs, conservation, sequence, reads, …)
  - download from public sources (UCSC, ENSEMBL), own annotations
- Flexibility
  - accommodate emerging data sources and structures
- Integration into existing Bioconductor landscape and standard data sources
- Scalability
- Structured API

Gviz package: implementation

- Loosely based on the existing GenomeGraphs package by James Bullard and Steffen Durinck
- Layout similar to the UCSC browser (tracks)
- Different data types are represented by different track classes
- Use efficient Bioconductor data structures (run-length encoding, optimized string representations) and vectorized graphics rendering
- Smart data summarization: only show the amount of detail according to the available plotting space
- Multitude of possible data inputs from within Bioconductor as well as from external sources (Ensembl, UCSC, flat files)
- Flexible settings API to control the look and feel

Track layout



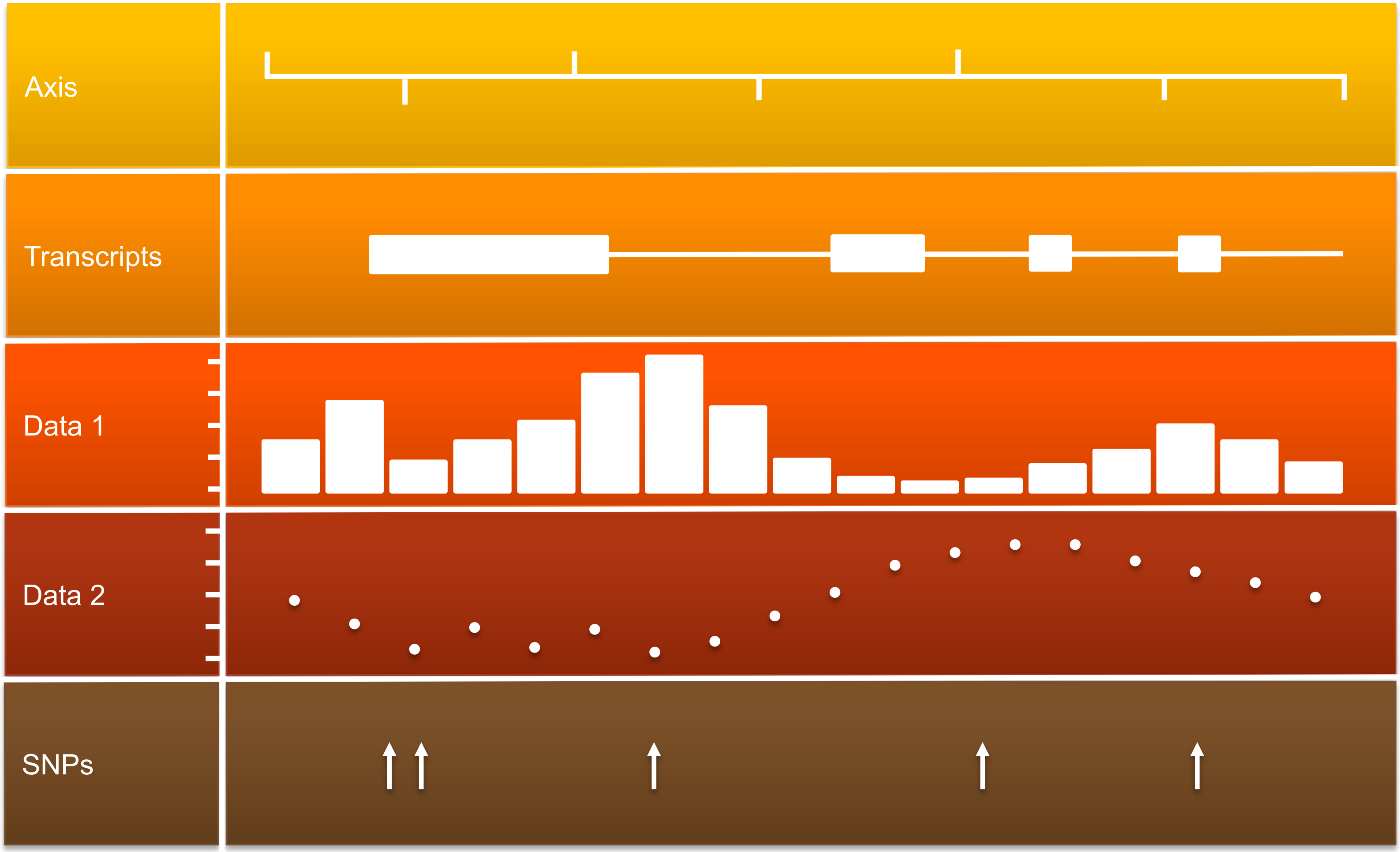
[Data range coordiantes]

Title Panel

Data Panel

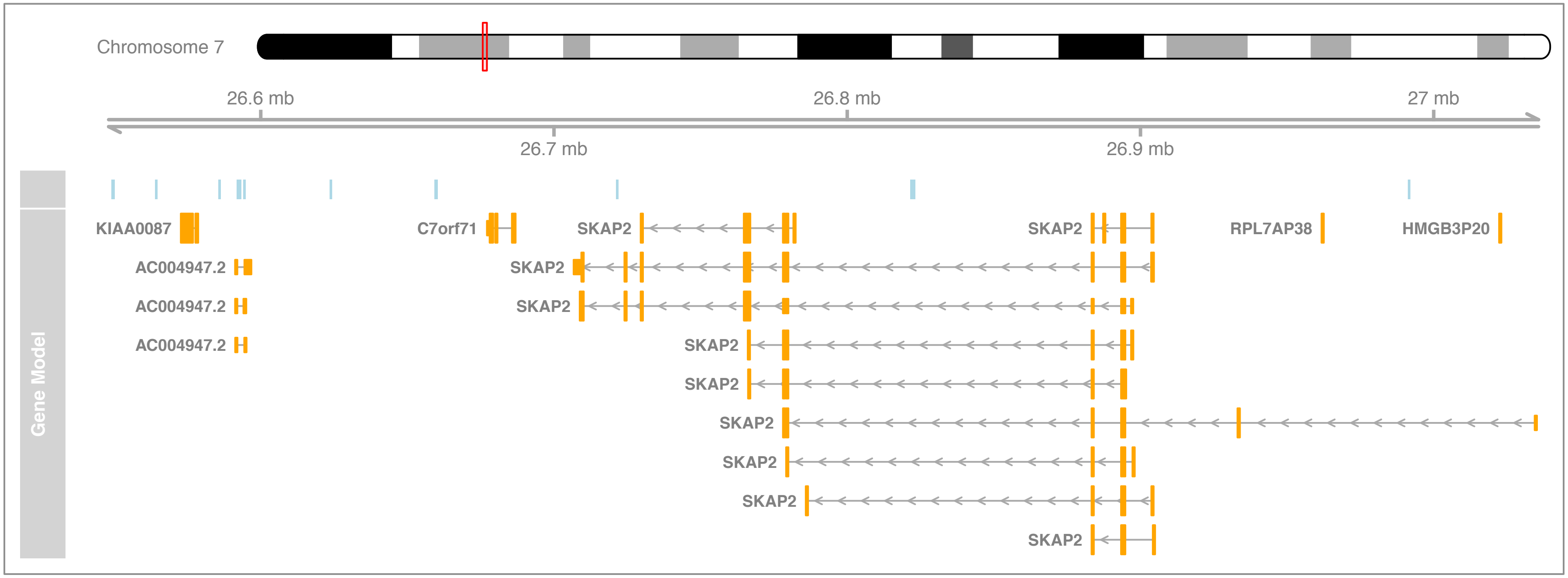
Genomic coordinates

Track panel layout



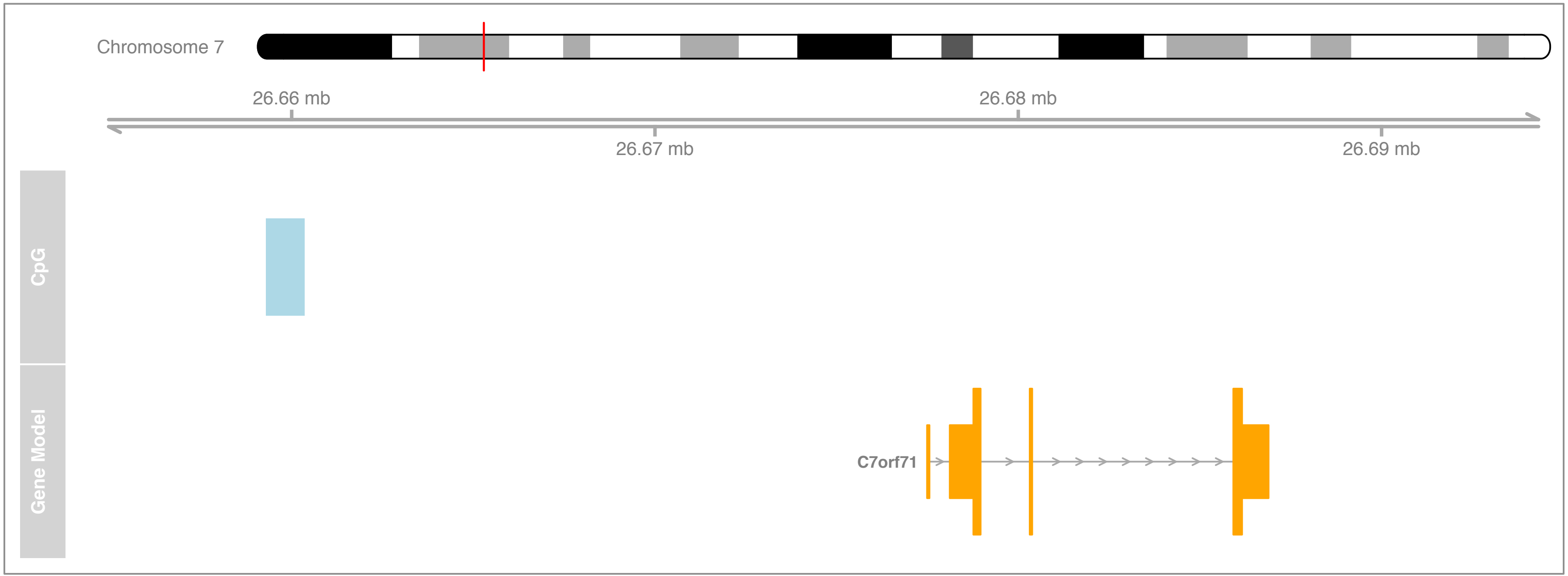
A typical Gviz session

```
> library(Gviz)
> data(cpgIslands)
> atr <- AnnotationTrack(cpgIslands, name="CpG")
> gtr <- GenomeAxisTrack()
> itr <- IdeogramTrack(genome="mm9", chromosome="chr1")
> data(geneModels)
> grtr <- GeneRegionTrack(geneModels, name="Gene Model", showId=TRUE)
> plotTracks(list(itr, gtr, atr, grtr))
```



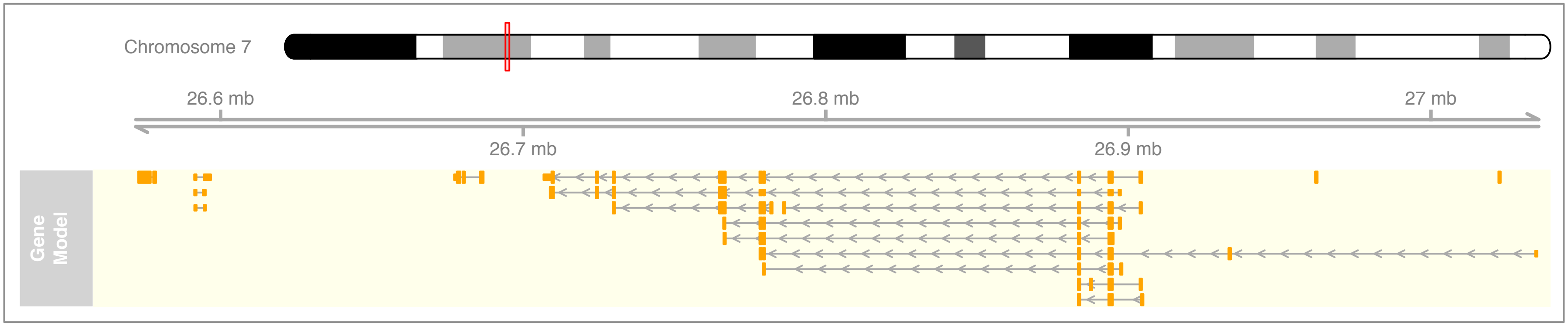
A typical Gviz session

```
> library(Gviz)
> data(cpgIslands)
> atr <- AnnotationTrack(cpgIslands, name="CpG")
> gtr <- GenomeAxisTrack()
> itr <- IdeogramTrack(genome="mm9", chromosome="chr1")
> data(geneModels)
> grtr <- GeneRegionTrack(geneModels, name="Gene Model", showId=TRUE)
> plotTracks(list(itr, gtr, atr, grtr), from=26654641, to=26694641)
```

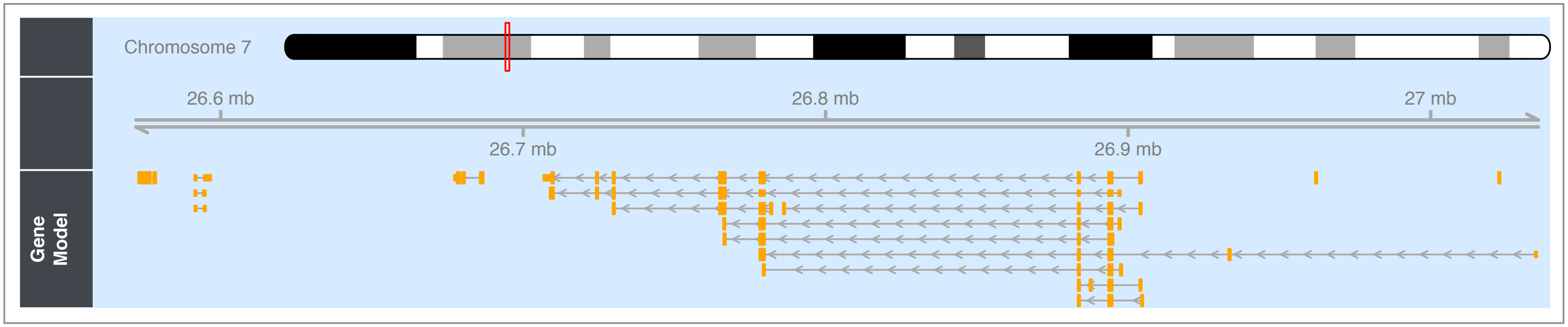


Display Parameters: controlling the look and feel

```
> displayPars(grtr) <- list(showId=FALSE, background.panel="#FFFFEB")
> plotTracks(list(itr, grtr))
```



```
> plotTracks(list(itr, grtr), background.panel="#D6EBFF”,
+    background.title="#40464C")
```



Display Parameters: documentation

- All available display parameters for a track class are documented in the “Display Parameters” section of the class documentation
- A list of available parameters along with their default settings can be shown using the `availableDisplayPars` function:

```
> head(availableDisplayPars("GeneRegionTrack"))
The following display parameters are available for 'GeneRegionTrack'
objects: (see ? GeneRegionTrack for details on their usage)

rotation (inherited from class 'AnnotationTrack'): 0
shape: smallArrow box
showAxis (inherited from class 'GdObject'): TRUE
showExonId: FALSE
showFeatureId (inherited from class 'AnnotationTrack'): FALSE
showId (inherited from class 'AnnotationTrack'): FALSE
showOverplotting (inherited from class 'AnnotationTrack'): FALSE
showTitle (inherited from class 'GdObject'): TRUE
size (inherited from class 'GdObject'): 1
```

Display Parameters: documentation

- All available display parameters for a track class are documented in the “Display Parameters” section of the class documentation

Display Parameters

The following display parameters are set for objects of class `GeneRegionTrack` upon instantiation, unless one or more of them have already been set by one of the optional sub-class initializers, which always get precedence over these global defaults. See `settings` for details on setting graphical parameters for tracks.

- `min.distance=0`: Numeric scalar. The minimum pixel distance before collapsing range items, only if \codecollapse==TRUE. See \code\linkcollapsing for details. Note that a value larger than 0 may lead to UTR regions being merged to CDS regions, which in most cases is not particularly useful.
- `col=NULL`: Character or integer scalar. The border color for all items. Defaults to using the same color as in `fill`, also taking into account different track `features`.
- `fill="orange"`: Character or integer scalar. The fill color for untyped items. This is also used to connect grouped items. See `grouping` for details.
- `geneSymbols=TRUE`: Logical scalar. Use human-readable gene symbols or gene IDs for the transcript annotation.
- `shape=c("smallArrow", "box")`: Character scalar. The shape in which to display the track items. Currently only `box`, `arrow`, `ellipse`, and `smallArrow` are implemented.
- `showExonId=FALSE`: Logical scalar. Control whether to plot the individual exon identifiers.
- `collapseTranscripts=FALSE`: Logical scalar. Merge all transcripts of the same gene into one single gene model. Essentially, this will only keep the start location of the first exon and the end location of the last exon from all transcripts of a gene.
- `thinBoxFeature=c("utr", "ncRNA", "utr3", "utr5", "miRNA", "lincRNA")`: Character vector. A listing of feature types that should be drawn with thin boxes. Typically those are non-coding elements.

Additional display parameters are being inherited from the respective parent classes. Note that not all of them may have an effect on the plotting of `GeneRegionTrack` objects.

- `AnnotationTrack`:
  - `cex=1`: Numeric scalar. The font expansion factor for item identifiers.
  - `cex.group=0.6`: Numeric scalar. The font expansion factor for the group-level annotation.
  - `col="transparent"`: Character or integer scalar. The border color for all track items.
  - `col.line="darkgray"`: Character scalar. The color used for connecting lines between grouped items. Defaults to a dark gray, but if set to `NULL` the same color as for the first item in the group is used.
  - `fontcolor="white"`: Character or integer scalar. The font color for item identifiers.
  - `fontcolor.group="#808080"`: Character or integer scalar. The font color for the group-level annotation.
  - `fontface=1`: Integer scalar. The font face for item identifiers.
  - `fontface.group=2`: Numeric scalar. The font face for the group-level annotation.
  - `fontfamily="sans"`: Character scalar. The font family for item identifiers.
  - `fontsize=12`: Numeric scalar. The font size for item identifiers.
  - `lex=1`: Numeric scalar. The line expansion factor for all track items. This is also used to connect grouped items. See `grouping` for details.
  - `lineheight=1`: Numeric scalar. The font line height for item identifiers.

```
>
T
C

r
s
s
s
s
s
s
showTitle (inherited from class 'GdObject'): TRUE
size (inherited from class 'GdObject'): 1
```

Display Parameters: documentation

- All available display parameters for a track class are documented in the “Display Parameters” section of the class documentation
- A list of available parameters along with their default settings can be shown using the `availableDisplayPars` function:
- The currently set display parameters of a track object can be queried using the `displayPars` function:

```
> head(displayPars(grtr))
$fill
[1] "orange”
$min.distance
[1] 0
$col
NULL
$geneSymbols
[1] TRUE
$showExonId
[1] FALSE
$collapseTranscripts
[1] FALSE
```

Track types: overview

Track	Description
GenomeAxisTrack	Genomic axis or scale indicator with optional highlighted regions.
IdeogramTrack	View of the displayed region on a schematic model of a chromosome with chromosome band information from UCSC
SequenceTrack	Genomic sequence in letter or false color representation depending on the zoom level.
AnnotationTrack	Generic annotation features (with at least start, stop, strand and chromosome information), optional grouping.
GeneRegionTrack	Gene or transcript models with grouping on the level of exons and transcripts. Can be fetched dynamically from Ensembl as the BiomartGeneRegionTrack child class.
DataTrack	Numeric values (single or grouped) along with genomic coordinates. Can be plotted in a variety of different ways.
AlignedReadsTrack	*Aligned NGS reads on the genome, either detailed view of individual reads or summarized coverage information.*

Track types: virtual parent classes

- Each track class inherits from `GdObject`:
  - general purpose methods and display parameters
- All track classes using genomic locations inherit from `RangeTrack`:
  - range-related methods and display parameters, range collapsing
- All track classes with potentially multiple lines in the output inherit from `StackedTrack`:
  - optimize stacking of items
- All track classes that contain associated numeric values inherit from `NumericTrack`:
  - setting up of y-axis, data aggregation
- All track classes that implement streaming from a file inherit from `ReferenceTrack`:
  - file handles and expected data structure definitions, streaming

GenomeAxisTrack: overview

Purpose:

Indicate the currently displayed genomic range either as a x-axis with evenly spaced tick marks or as a scale reference.

Inputs:

NA

Details:

- Ranges on the axis can be highlighted, e.g. to indicate stretches of N nucleotides

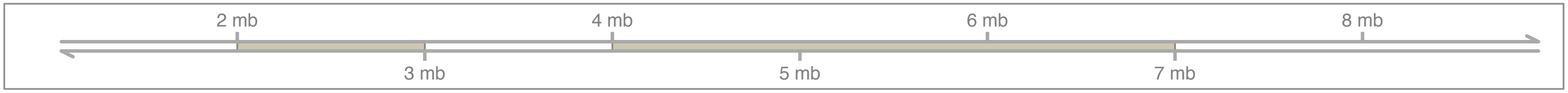
GenomeAxisTrack: examples

Default axis:

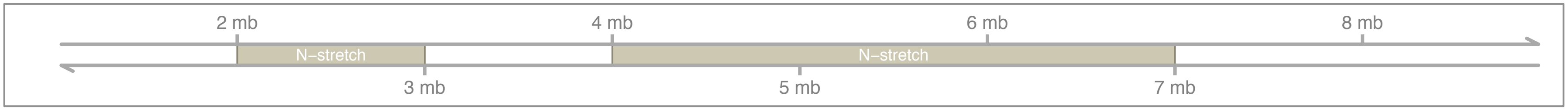
```
> axisTrack <- GenomeAxisTrack()
> plotTracks(axisTrack, from=1e06, to=9e6)
```

Highlighting ranges on the axis:

```
> axisTrack <- GenomeAxisTrack(range=IRanges(start=c(2e6, 4e6),
+     end=c(3e6, 7e6), names=rep("N-stretch", 2)))
> plotTracks(axisTrack, from=1e6, to=9e6)
```



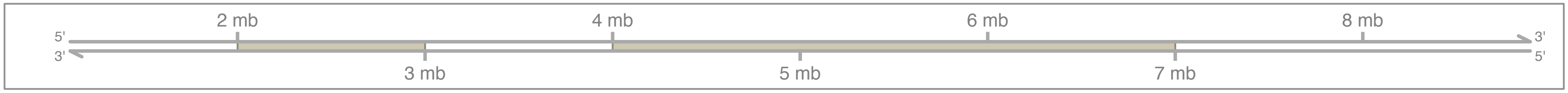
```
> plotTracks(axisTrack, from=1e6, to=9e6, showId=TRUE)
```



GenomeAxisTrack: examples

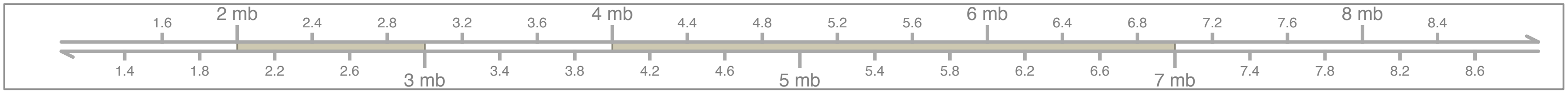
Direction indicators:

```
> plotTracks(axisTrack, from=1e6, to=9e6, add53=TRUE, add35=TRUE)
```

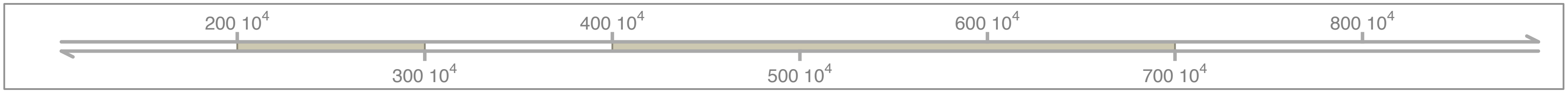


Tick mark formatting:

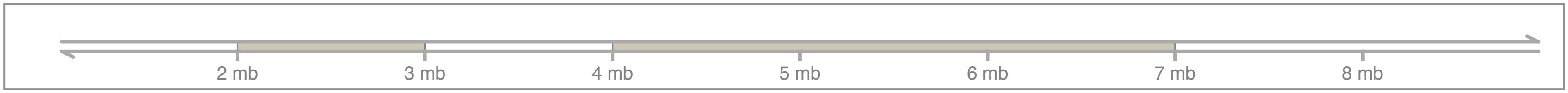
```
> plotTracks(axisTrack, from=1e6, to=9e6, littleTicks=TRUE)
```



```
> plotTracks(axisTrack, from=1e6, to=9e6, exponent=4)
```



```
> plotTracks(axisTrack, from=1e6, to=9e6, labelPos="below")
```



GenomeAxisTrack: examples

Scale reference (relative size):

```
> plotTracks(axisTrack, from=1e6, to=9e6, scale=0.5)
```

Scale reference (absolute size):

```
> plotTracks(axisTrack, from=1e6, to=9e6, scale=1e6)
```

IdeogramTrack: overview

Purpose:

Indicate the currently displayed genomic range in the context of the current chromosome.

Inputs:

- Fetch chromosome band information for the genome from UCSC
- `data.frame`

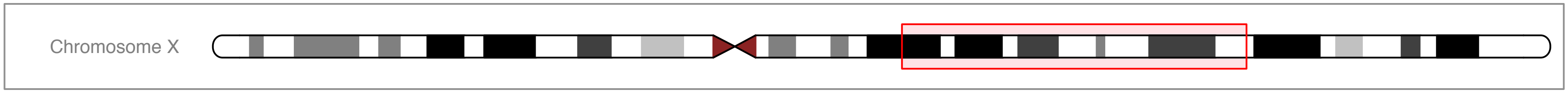
Details:

- After the first connection to UCSC the fetched results are being cached for the duration of the R session

IdeogramTrack: examples

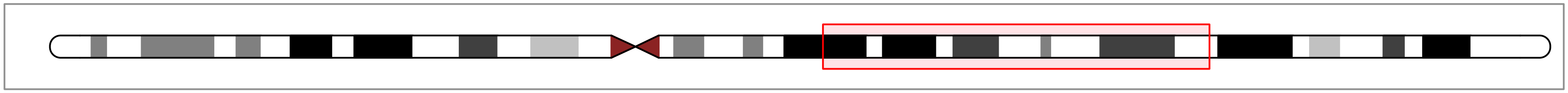
Default ideogram:

```
> ideoTrack <- IdeogramTrack(genome="hg19", chromosome="chrX")
> plotTracks(ideoTrack, from=8e7, to=12e7)
```



Chromosome name:

```
> plotTracks(ideoTrack, from=8e7, to=12e7, showId=FALSE)
```



Chromosome band identifiers:

```
> plotTracks(ideoTrack, from=8e7, to=12e7, showId=FALSE, showBandId=TRUE)
```

SequenceTrack: overview

Purpose:

Show genomic sequence of the currently displayed region

Inputs:

- `DNAStringSet`
- `BSgenome`
- `FASTA` file (indexed or not indexed)
- `2bit` file

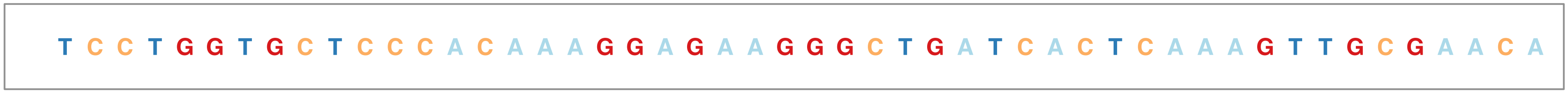
Details:

- Depending on the zoom level, sequences will be shown as individual letters, as color-coded boxes or as a horizontal line

SequenceTrack: examples

Default sequence track:

```
> library(BSgenome.Hsapiens.UCSC.hg19)
> sTrack <- SequenceTrack(Hsapiens)
> plotTracks(sTrack, chromosome="chr1", from=20000, to=20050)
```



Direction indicators:

```
> plotTracks(sTrack, chromosome=1, from=20000, to=20050, add53=TRUE)
```

Sequence complement:

```
> plotTracks(sTrack, chromosome=1, from=20000, to=20050, add53=TRUE,
+     complement=TRUE)
```

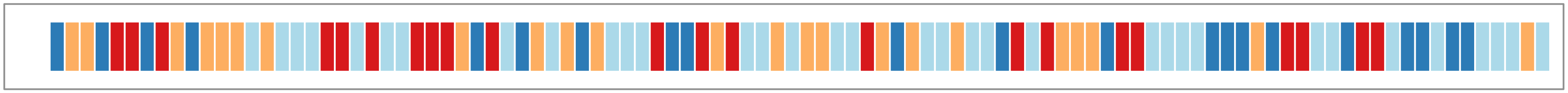
SequenceTrack: examples

False color coding of letters:

```
> fcol <- c(A="darkgray", C="darkgray", T="darkgray", G="darkgray")
> plotTracks(sTrack, chromosome=1, from=20000, to=20050, fontcolor=fcol)
```

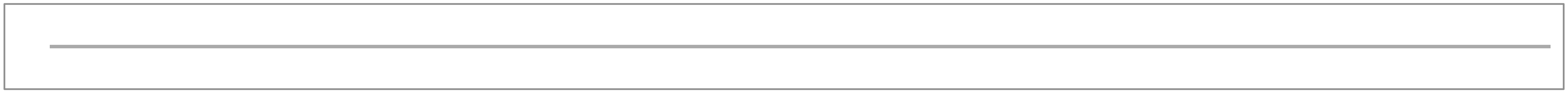
Zoom level details:

```
> plotTracks(sTrack, chromosome=1, from=20000, to=20100)
```



```
> plotTracks(sTrack, chromosome=1, from=20000, to=20100, cex=0.5)
```

```
> plotTracks(sTrack, chromosome=1, from=20000, to=201000)
```



AnnotationTrack: overview

Purpose:

Simple annotation features with at least start, stop, strand and chromosome information. Items can be grouped and colored according to type.

Inputs:

- `IRanges` (+ chromosome and strand as separate arguments)
- `GRanges`
- `GRangesList`
- Various file types: e.g., `BED`, `GFF`, `BAM`

Details:

- Overlapping items are stacked for optimal utilization of available plotting space
- Depending on the available space and resolution some items may be merged
- Additional information for each annotation item can be added by means of the `DetailsAnnotationTrack` child class

AnnotationTrack: examples

Default annotation track:

```
> aTrack <- AnnotationTrack(start=c(10,40,120), width=15, chromosome="chrX",
+     strand=c("+", "*", "-"), id=c("Huey", "Dewey", "Louie"),
+     genome="hg19", name="foo")
> plotTracks(aTrack)
```

Feature shapes:

```
> plotTracks(aTrack, shape="box", showFeatureId=TRUE)
```

```
> plotTracks(aTrack, shape="ellipse", showFeatureId=TRUE, fontcolor="blue")
```

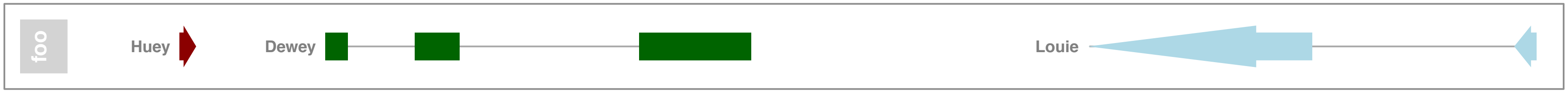
AnnotationTrack: examples

Grouped features:

```
> aTrack.groups <- AnnotationTrack(start=c(50,180,260,460,860,1240),
+     width = c(15,20,40,100,200,20), chromosome="chrX",
+     strand=rep(c("+","*", "-"), c(1, 3, 2)), group = rep(c("Huey",
+     "Dewey”,"Louie"), c(1,3,2)), genome="hg19", name="foo")
> plotTracks(aTrack.groups, showId=TRUE)
```

Feature type color coding:

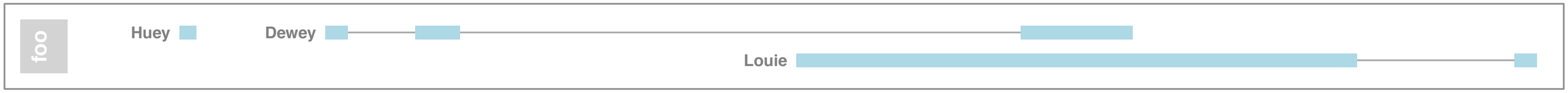
```
> feature(aTrack.groups)[1:4] <- c("foo","bar","bar","bar")
> plotTracks(aTrack.groups, showId=TRUE, foo="darkred", bar="darkgreen")
```



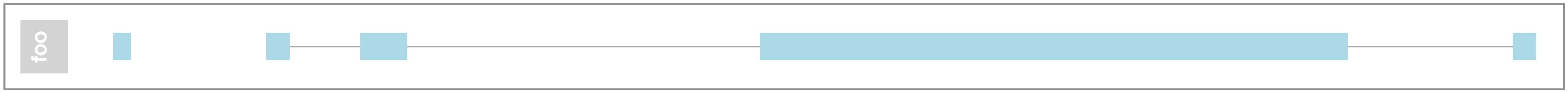
AnnotationTrack: examples

Feature stacking:

```
> aTrack.stacked <- AnnotationTrack(start=c(50,180,260,800,600,1240),
+     width=c(15,20,40,100,500,20), chromosome="chrX", strand="*",name="foo"
+     group=rep(c("Huey","Dewey","Louie"), c(1,3,2)), genome="hg19")
> plotTracks(aTrack.stacked, showId=TRUE)
```

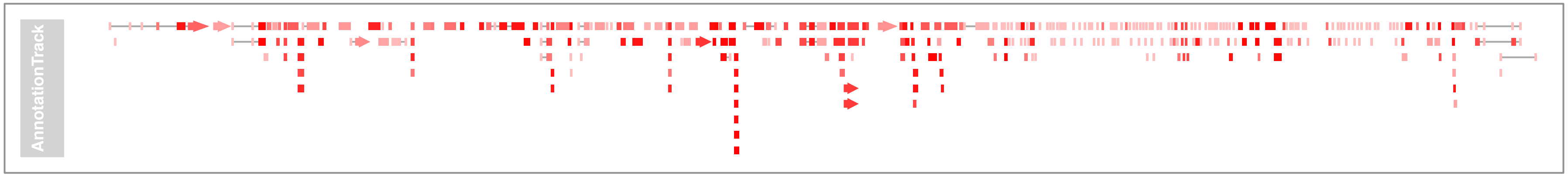


```
> plotTracks(    aTrack.stacked, stacking="dense")
```



Overplotting density:

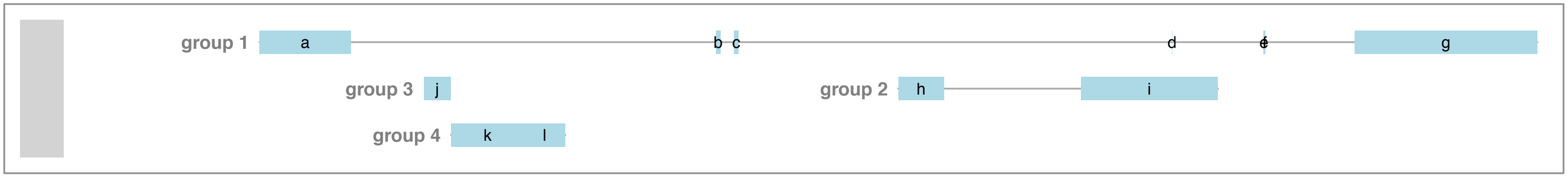
```
> data("denseAnnTrack")
> plotTracks(denseAnnTrack, showOverplotting=TRUE)
```



AnnotationTrack: item collapsing details

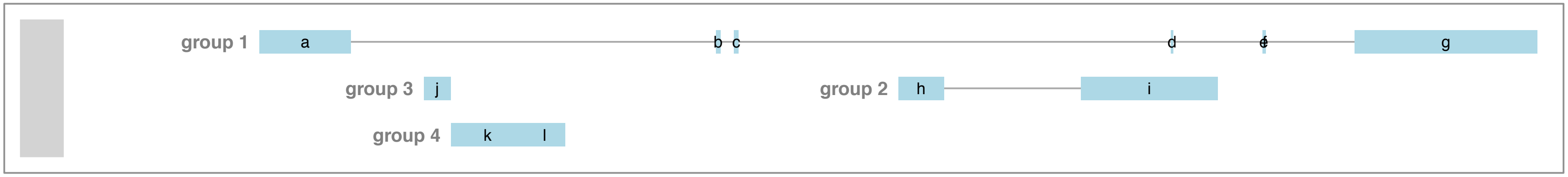
All item collapsing and width expansion disabled:

```
> data(collapseTrack)
> plotTracks(ctrack)
```



The min.width setting:

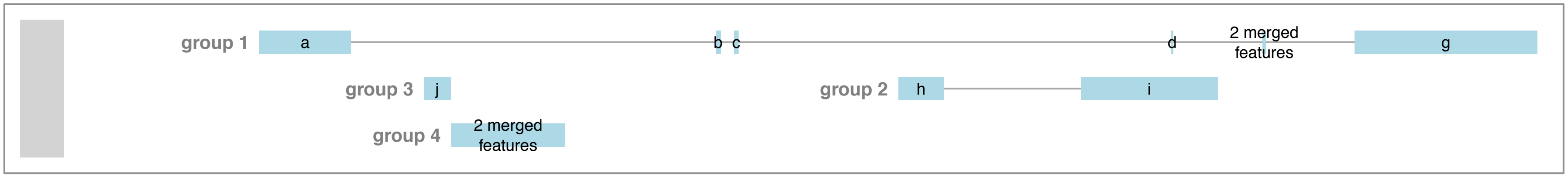
```
plotTracks(ctrack, min.width=1)
```



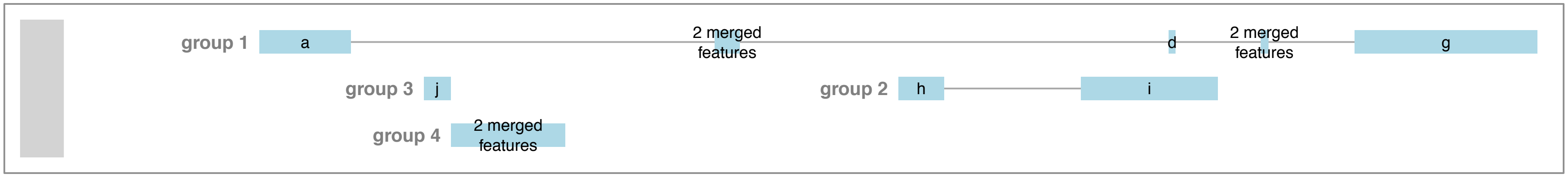
AnnotationTrack: item collapsing details

The min.width and min.distance settings and item collapsing:

```
plotTracks(ctrack, min.width=1, collapse=TRUE)
```

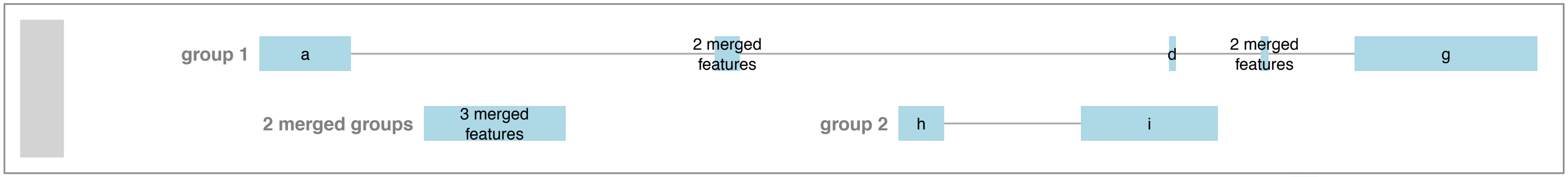


```
> plotTracks(ctrack, min.width=3, min.distance=5, collapse=TRUE)
```



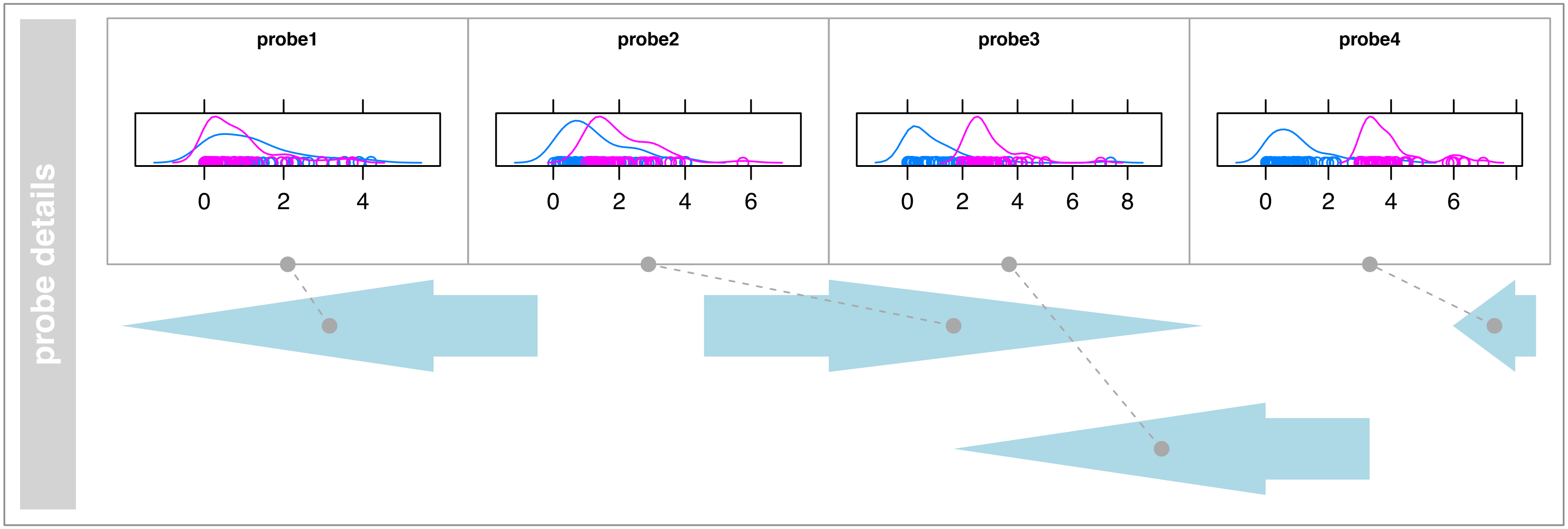
The mergeGroups setting:

```
> plotTracks(ctrack, min.width=3, min.distance=5, collapse=TRUE,
+      mergeGroups=TRUE)
```



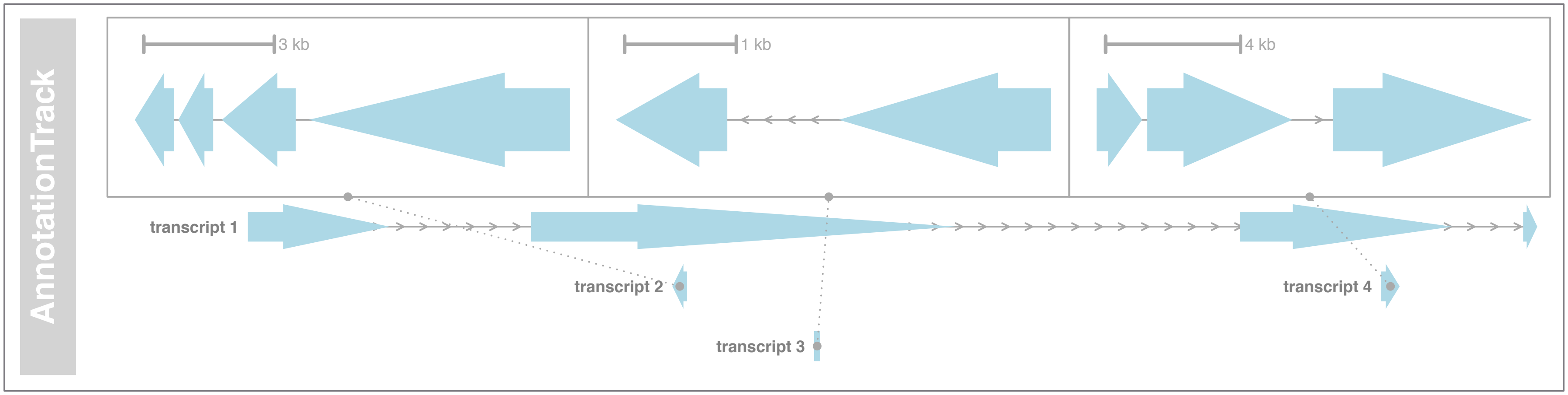
AnnotationTrack: DetailsAnnotationTrack child class

```
> details <- function(identifier, ...) {
+     d <- data.frame(signal=methylation[identifier,], group=sgroups)
+     print(densityplot(~signal, group=group, data=d,
+     main=list(label=identifier, cex=0.7), scales=list(draw=FALSE,
+     x=list(draw=TRUE)), ylab="", xlab="", ), newpage=FALSE,
+     prefix="plot")}
> deTrack <- AnnotationTrack(range = probes, genome = "hg19",
+     chromosome=7, id=rownames(methylation), name="probe details",
+     stacking="squish", fun=details)
> plotTracks(deTrack)
```



AnnotationTrack: DetailsAnnotationTrack child class

```
selFun <- function(identifier, start, end, track, GdObject, ...) {
+     gcount <- table(group(GdObject))
+     pxRange <- Gviz:::.pxResolution(min.width = 20, coord="x")
+     return((end - start) < pxRange && gcount[identifier] == 1)}
detFun <- function(identifier, GdObject.original, ...) {
+     plotTracks(list(GenomeAxisTrack(scale=0.3, size=0.2, cex=0.7),
+     GdObject.original[group(GdObject.original) == identifier]),
+     add=TRUE, showTitle=FALSE)}
deTrack2 <- AnnotationTrack(geneDetails, fun=detFun, selectFun=selFun,
+     groupDetails=TRUE, details.size=0.5, detailsConnector.cex=0.5,
+     detailsConnector.lty="dotted", shape=c("smallArrow", "arrow"),
+     showId=TRUE)
 plotTracks(deTrack2)
```



GeneRegionTrack: overview

Purpose:

Gene model annotations.

Inputs:

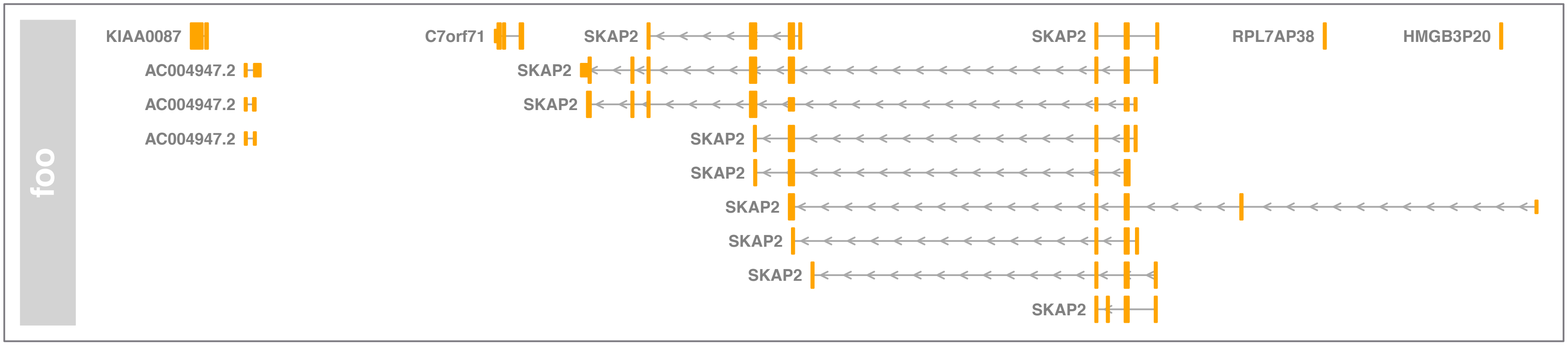
- `IRanges` (+ chromosome and strand as separate arguments)
- `GRanges`
- `GRangesList`
- `TranscriptDb`
- Various file types: e.g., `GFF`, `GTF`
- Direct import from Ensembl via the biomaRt interface

Details:

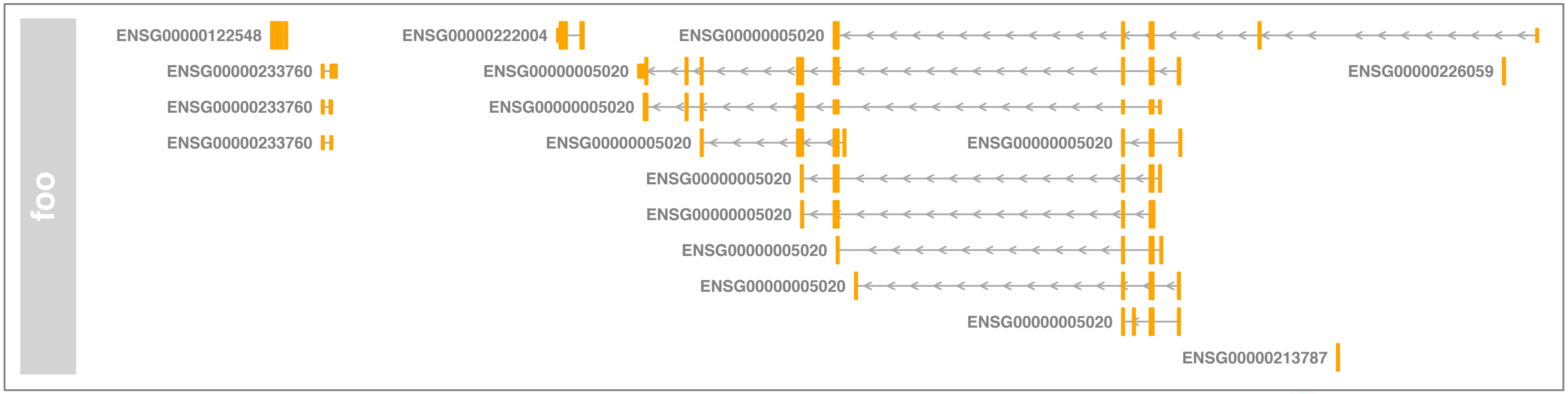
- Modeling of exon, transcript and gene relationships
- Support for human-readable gene symbols
- Support for coding and non-coding elements

GeneRegionTrack: gene symbols and gene ids

```
> data(geneModels)
> grtrack <- GeneRegionTrack(geneModels, genome="hg19", name="foo")
> plotTracks(grtrack, showId=TRUE)
```



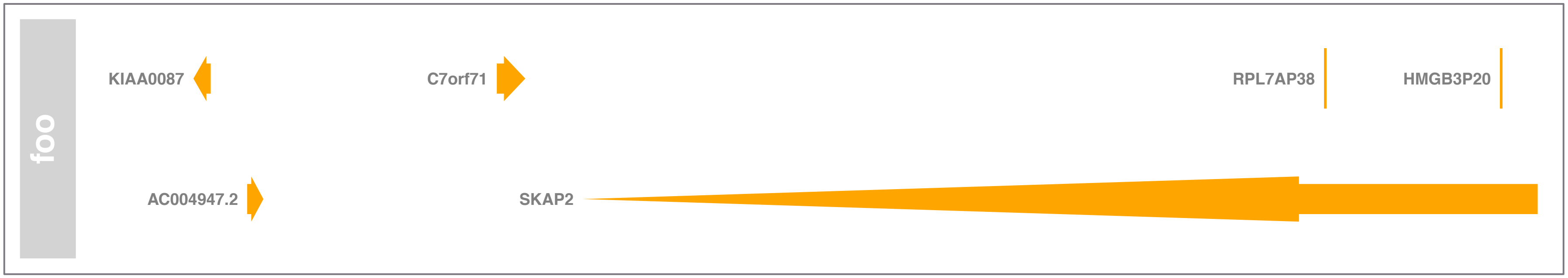
```
> plotTracks(grtrack, showId=TRUE, geneSymbols=FALSE)
```



GeneRegionTrack: examples

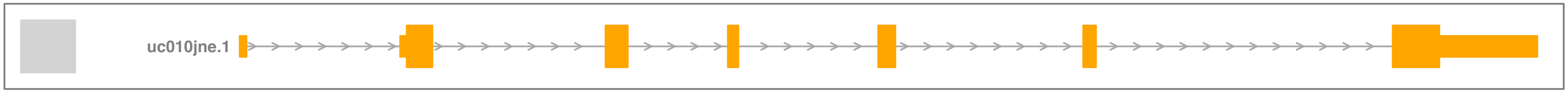
Collapsing exons information:

```
> plotTracks(grtrack, collapseTranscripts=TRUE, shape="arrow", showId=TRUE)
```



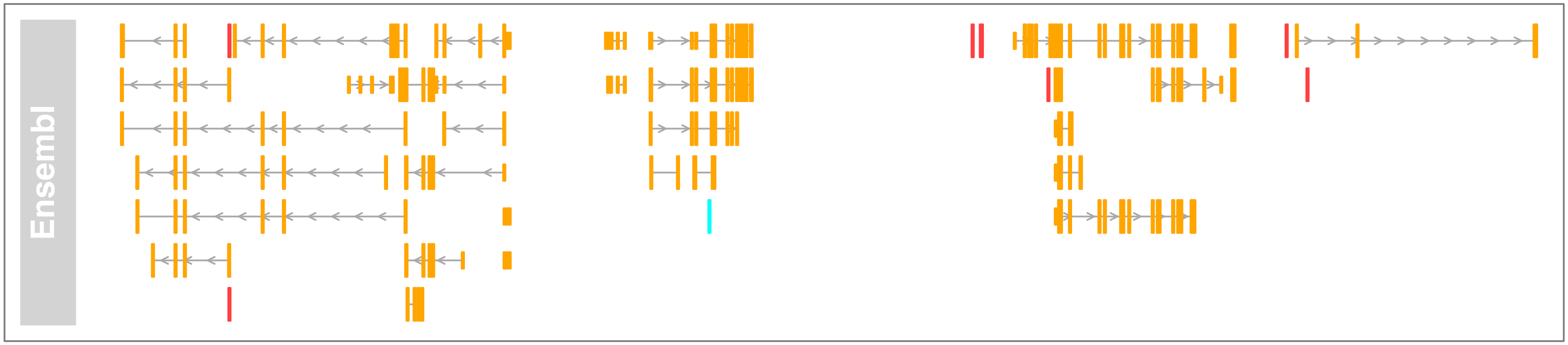
Coding and non-coding regions

```
>  txdb <- loadDb(system.file("extdata", "UCSC_knownGene_sample.sqlite",
+     package="GenomicFeatures"))
> txTr <- GeneRegionTrack(txdb, chromosome="chr6", start=300000, end=350000)
> plotTracks(txTr, showId=TRUE)
```

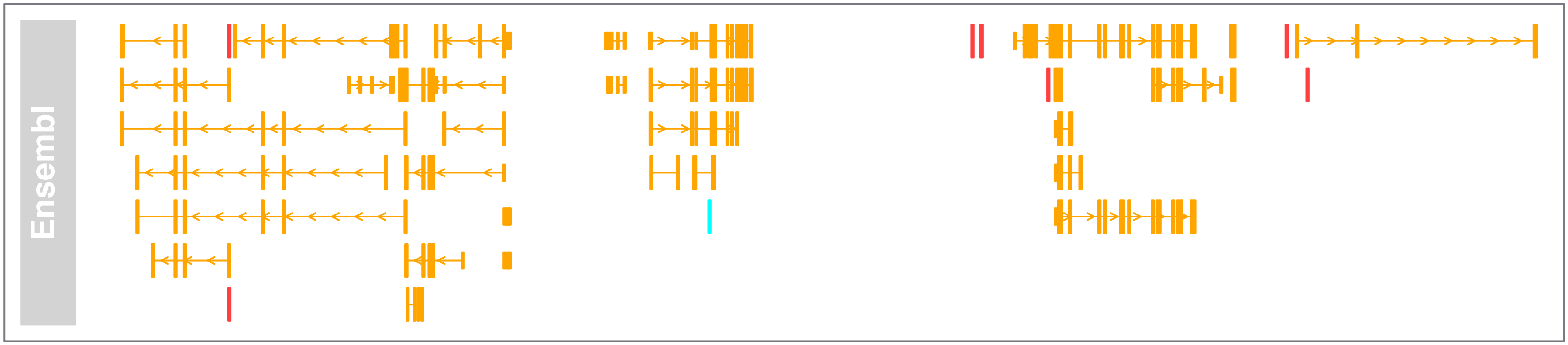


GeneRegionTrack: BiomartGeneRegionTrack

```
> biomTrack <- BiomartGeneRegionTrack(genome="hg19", chromosome="chr7",
+      start=2e7, end=2.1e7, name="Ensembl")
> plotTracks(biomTrack)
```



```
> plotTracks(biomTrack, col.line=NULL)
```



DataTrack: overview

Purpose:

Numeric data along genomic coordinates.

Inputs:

- `IRanges` (+ chromosome, strand and data matrix)
- `GRanges`
- Various file types: e.g., `WIG`, `BedGraph`, `BigWig`, `BAM`

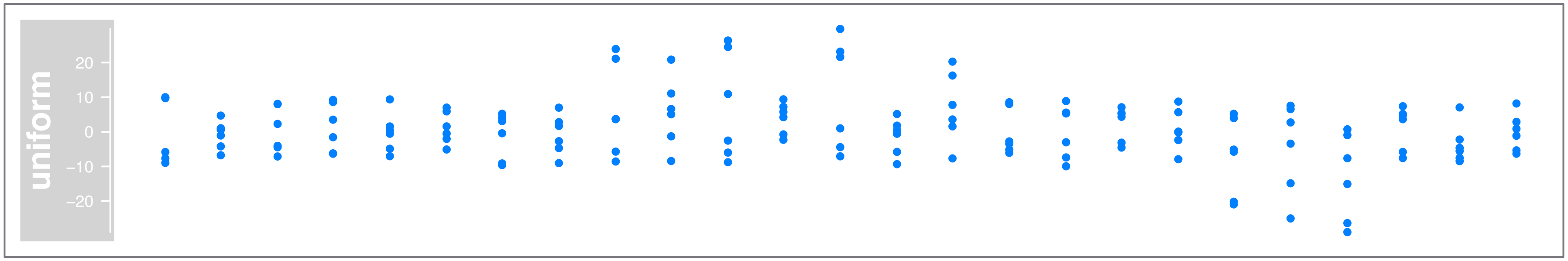
Details:

- Flexible visualization options (e.g. line charts, scatter plots, box plots, bar charts)
- Sample grouping
- Data transformations

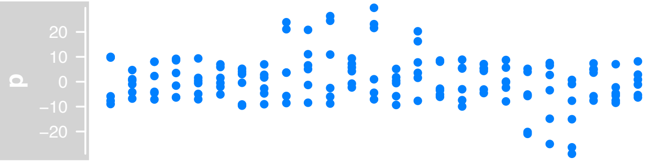
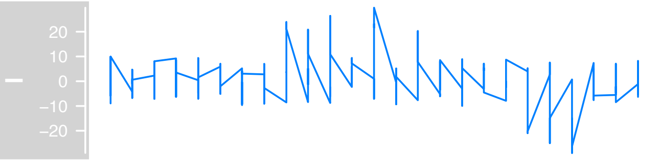
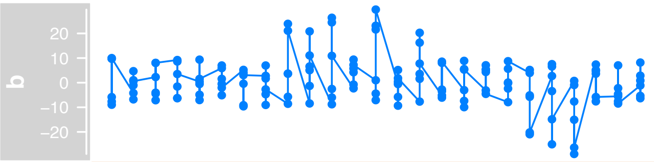
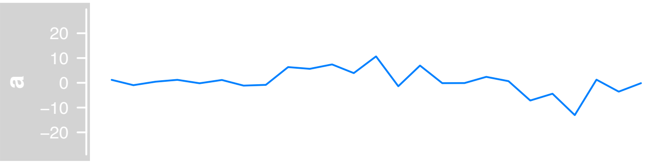
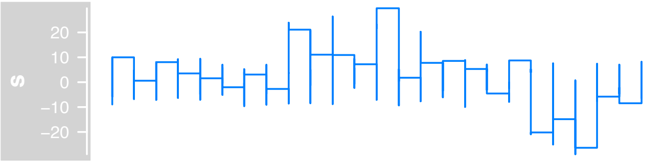
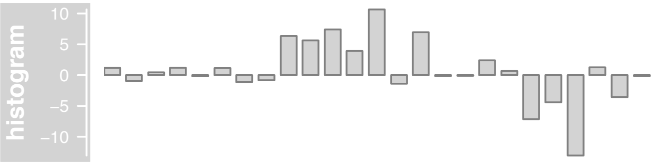
DataTrack: examples

Default data track:

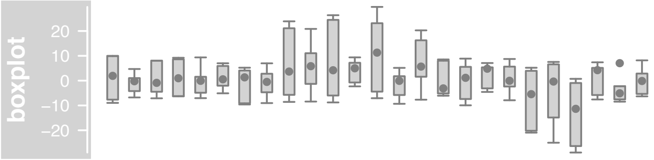
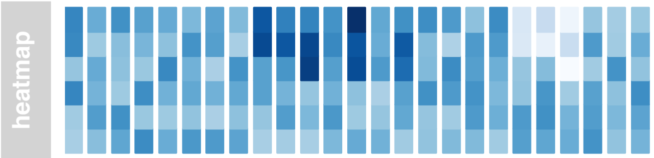
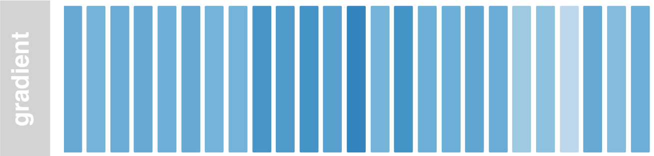
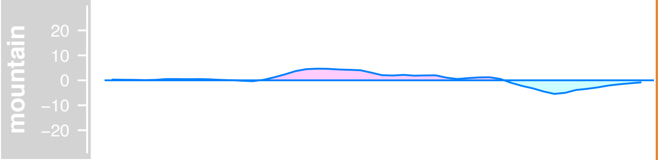
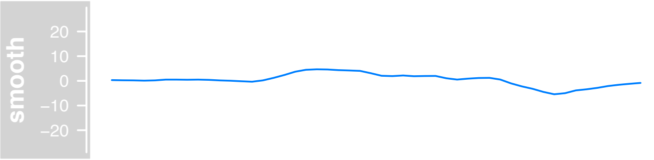
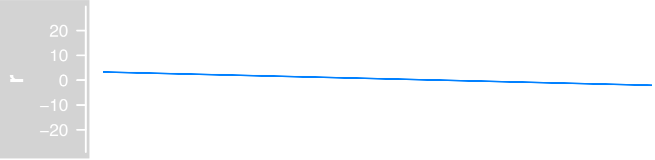
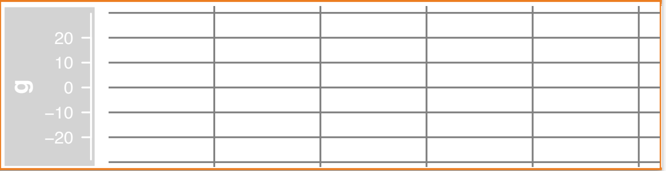
```
> data(twoGroups)
> dim(mcols(twoGroups))
[1] 25  6
> dTrack <- DataTrack(twoGroups, name="uniform")
> plotTracks(dTrack)
```



DataTrack: plot types

Type	Description	Example
p	Dot plot	
l	Line charts	
b	Dot and line charts	
a	Line chart of average values	
s/S	Horizontal first or vertical first stair steps	
h	Lines barchart	
histogram	Bar chart, bar width equal to range width	

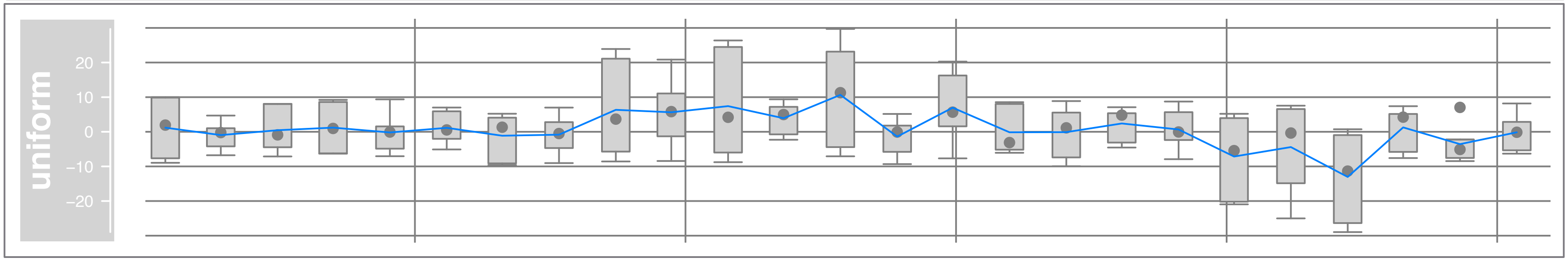
DataTrack: plot types

Type	Description	Example
boxplot	Box-and whiskers plot	
heatmap	False color image of individual values	
gradient	False color image of summarized values	
mountain	Mountain-type plot relative to a baseline	
smooth	Loess curve	
r	Linear regression line	
g	Grid lines	

DataTrack: examples

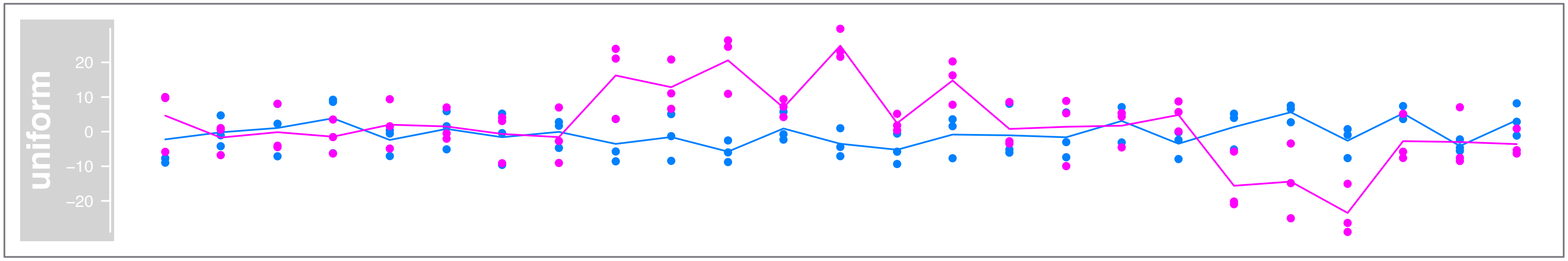
Combining plot types:

```
> plotTracks(dTrack, type = c("boxplot", "a", "g"))
```



Sample grouping:

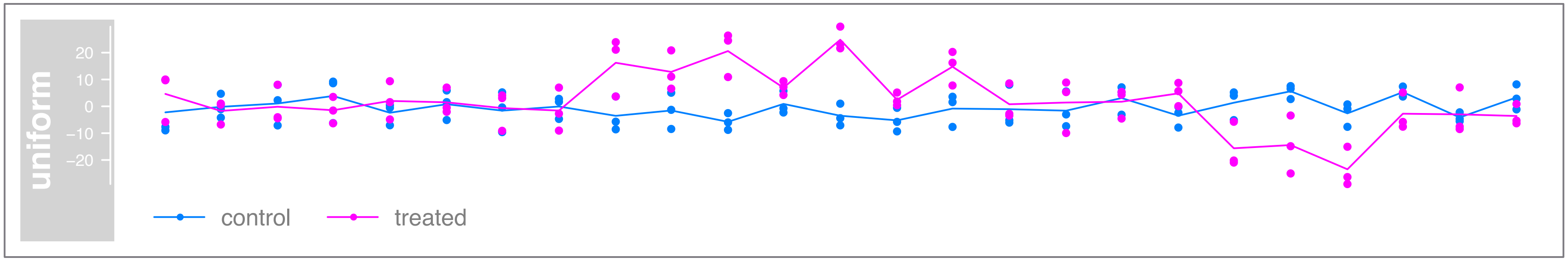
```
> plotTracks(dTrack, groups=rep(c("control", "treated"), each=3),
+      type=c("a","p"))
```



DataTrack: examples

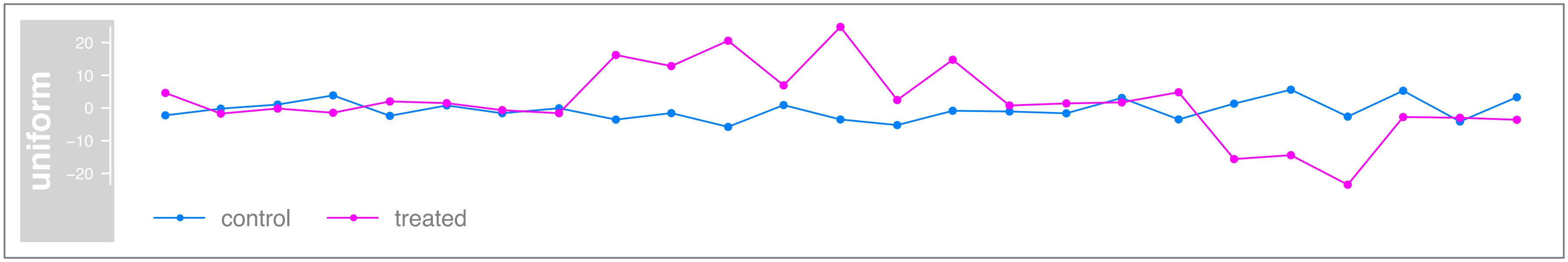
Sample groups legend:

```
> plotTracks(dTrack, groups=rep(c("control", "treated"), each=3),
+      type=c("a", "p"), legend=TRUE)
```



Within-group data aggregation:

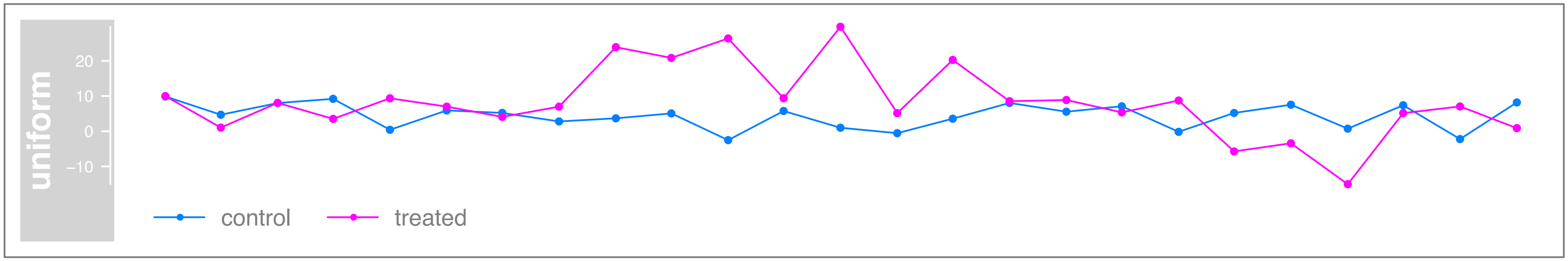
```
> plotTracks(dTrack, groups=rep(c("control", "treated"), each=3),
+      type=c("a","p"), legend=TRUE, aggregateGroups=TRUE)
```



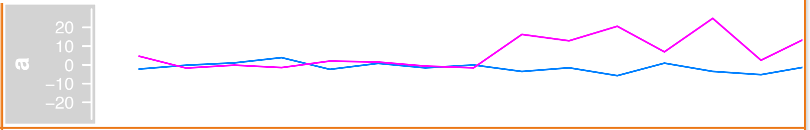
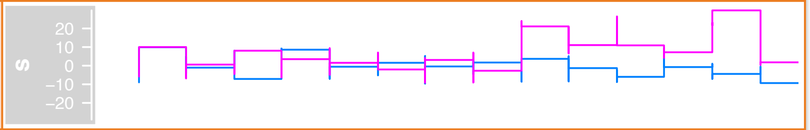
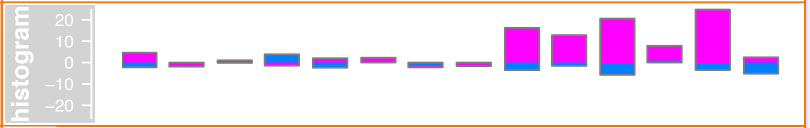
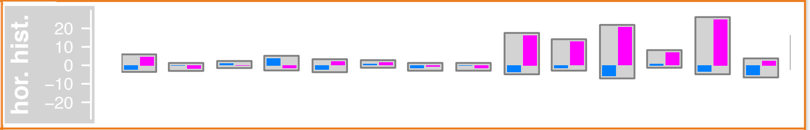
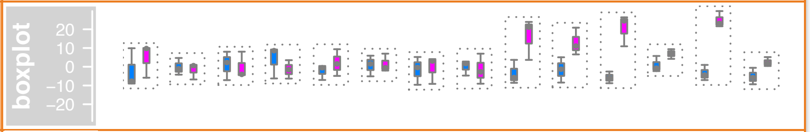
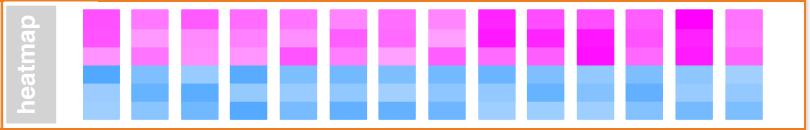
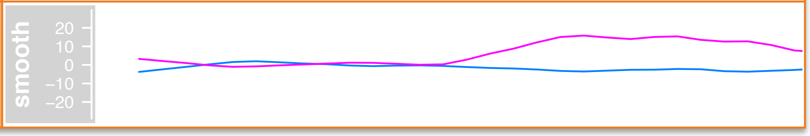
DataTrack: examples

Data aggregation method:

```
> plotTracks(dTrack, groups=rep(c("control", "treated"), each=3),
+      type=c("a","p"), legend=TRUE, aggregateGroups=TRUE, aggregation="max")
```



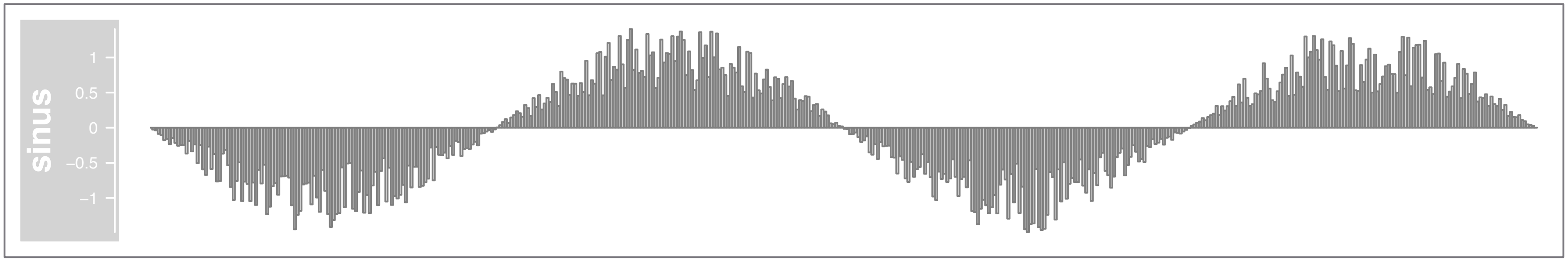
DataTrack: grouped plot types

Type	Description	Example
a	Line chart of group average values	
s/S	Horizontal first or vertical first stair steps	
histogram	Stacked bar chart	
horizontal histogram	Side by side bar chart	
boxplot	Box-and-whiskers plots stratified by group	
heatmap	False color image of average group values	
smooth	Loess curves stratified by group	

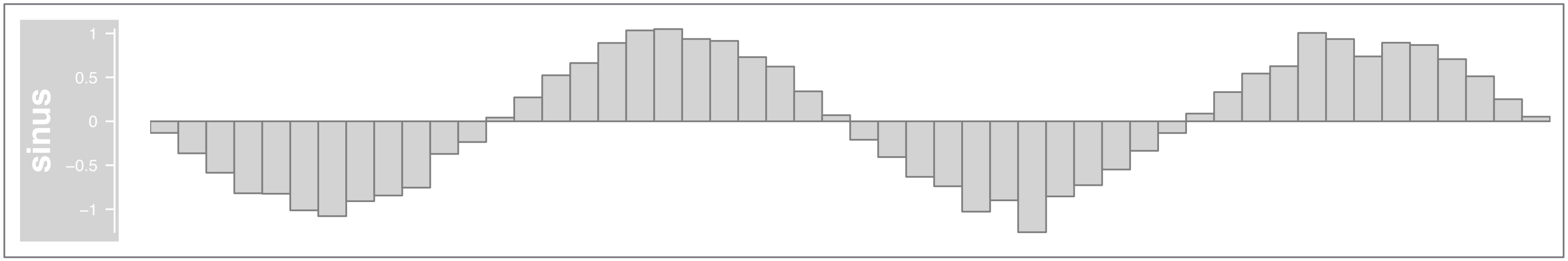
DataTrack: transformations

Fixed window operations:

```
> dat <- sin(seq(pi, 10 * pi, len=500))
> dTrack.big <- DataTrack(start=seq(1, 1e5, len=500), width=15,
+     chromosome="chrX", genome="hg19", name="sinus",
+     data=sin(seq(pi, 5*pi, len=500)) * runif(500, 0.5, 1.5))
> plotTracks(dTrack.big, type="histogram")
```



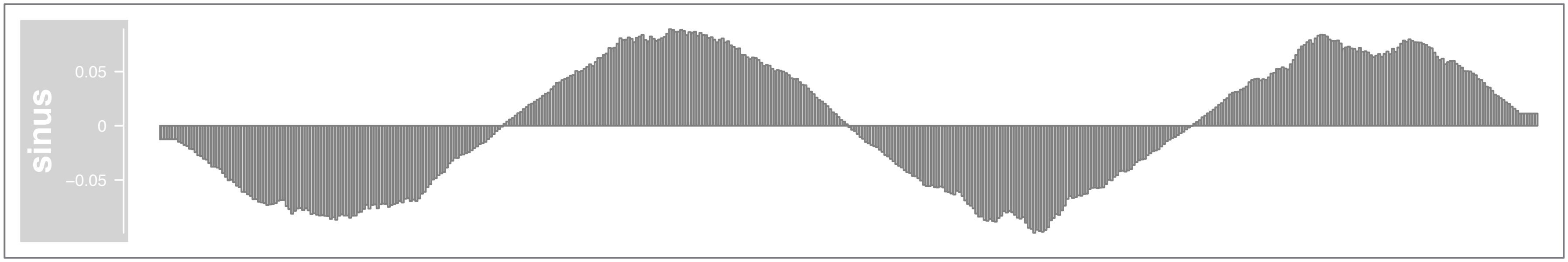
```
> plotTracks(dTrack.big, type="histogram", window=50)
```



Track types: DataTrack

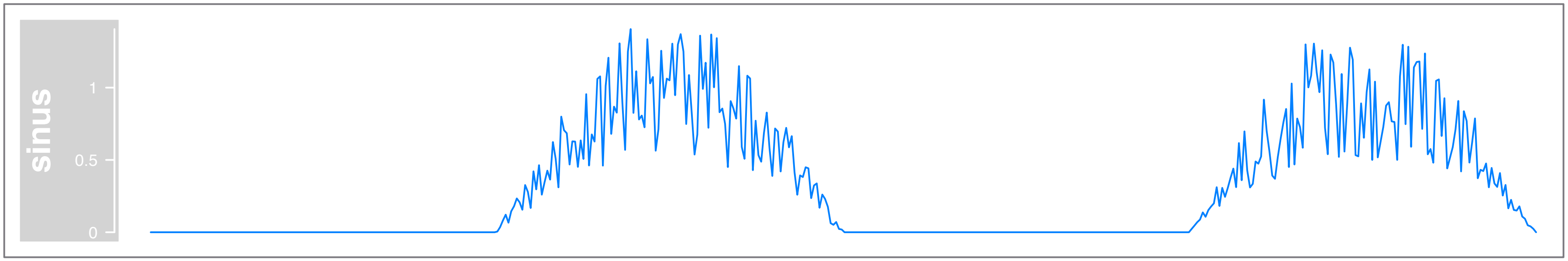
Sliding window operations:

```
> plotTracks(dTrack.big, type="histogram", window=-1, windowSize=2500)
```

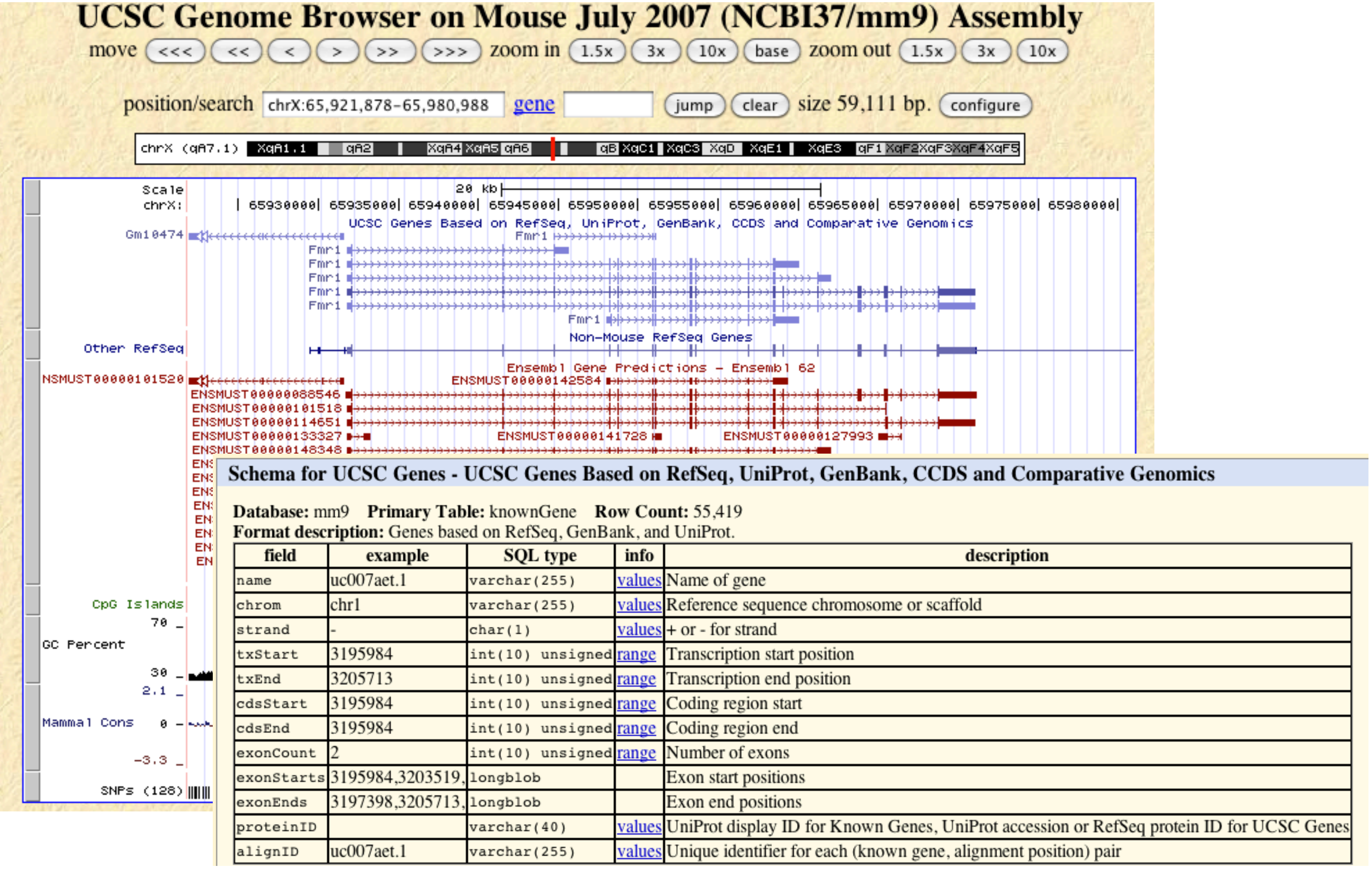


Transformation operations:

```
> tfun <- function(x) ifelse(x < 0, 0, x)
> plotTracks(dTrack.big, type="l", transformation=tfun)
```



Fetching data directly from UCSC



UCSC Genome Browser on Mouse July 2007 (NCBI37/mm9) Assembly

position/search chrX:65,921,878-65,980,988 size 59,111 bp.

Schema for UCSC Genes - UCSC Genes Based on RefSeq, UniProt, GenBank, CCDS and Comparative Genomics

Database: mm9 **Primary Table:** knownGene **Row Count:** 55,419
Format description: Genes based on RefSeq, GenBank, and UniProt.

field	example	SQL type	info	description
name	uc007aet.1	varchar(255)	values	Name of gene
chrom	chr1	varchar(255)	values	Reference sequence chromosome or scaffold
strand	-	char(1)	values	+ or - for strand
txStart	3195984	int(10) unsigned	range	Transcription start position
txEnd	3205713	int(10) unsigned	range	Transcription end position
cdsStart	3195984	int(10) unsigned	range	Coding region start
cdsEnd	3195984	int(10) unsigned	range	Coding region end
exonCount	2	int(10) unsigned	range	Number of exons
exonStarts	3195984,3203519,	longblob		Exon start positions
exonEnds	3197398,3205713,	longblob		Exon end positions
proteinID		varchar(40)	values	UniProt display ID for Known Genes, UniProt accession or RefSeq protein ID for UCSC Genes
alignID	uc007aet.1	varchar(255)	values	Unique identifier for each (known gene, alignment position) pair

Fetch data directly from UCSC

Simple annotation tracks:

```
> from <- 65921878
> to <- 65980988
> cpgIslands <- UcscTrack(genome="mm9", chromosome="chrX",
+     track="cpgIslandExt", from=from, to=to,
+     trackType="AnnotationTrack", start="chromStart", end="chromEnd",
+     id="name", shape="box", fill="#006400", name="CpG Islands")
```

```
> snpLocations <- UcscTrack(genome="mm9", chromosome="chrX",
+     track="snp128", from=from, to=to, trackType="AnnotationTrack",
+     start="chromStart", end="chromEnd", id="name", feature="func",
+     strand="strand", shape="box", stacking="dense", fill="black",
+     name="SNPs")
```

Fetching data directly from UCSC

Gene model type tracks:

```
> knownGenes <- UcscTrack(genome="mm9", chromosome="chrX",
+     track="knownGene", from=from, to=to, trackType="GeneRegionTrack",
+     rstarts="exonStarts", rends="exonEnds", gene="name", symbol="name",
+     transcript="name", strand="strand", fill="#8282d2", name="UCSC Genes")
```

```
> refGenes <- UcscTrack(genome="mm9", chromosome="chrX",
+     track="xenoRefGene", from=from, to=to,
+     trackType="GeneRegionTrack", rstarts="exonStarts",
+     rends="exonEnds", gene="name", symbol="name2",
+     transcript="name", strand="strand", fill="#8282d2",
+     stacking="dense", name="Other RefSeq")
```

```
> ensGenes <- UcscTrack(genome="mm9", chromosome="chrX",
+     track="ensGene", from=from, to=to, trackType="GeneRegionTrack",
+     rstarts="exonStarts", rends="exonEnds", gene="name",
+     symbol="name2", transcript="name", strand="strand",
+     fill="#960000", name="Ensembl Genes")
```

Fetching data directly from UCSC

Data tracks:

```
> conservation <- UcscTrack(genome="mm9", chromosome="chrX",
+     track="Conservation", table="phyloP30wayPlacental", from=from, to=to,
+     trackType="DataTrack", start="start", end="end", data="score",
+     type="hist", window="auto", col.histogram="darkblue",
+     fill.histogram="darkblue", ylim=c(-3.7, 4), name="Conservation")
```

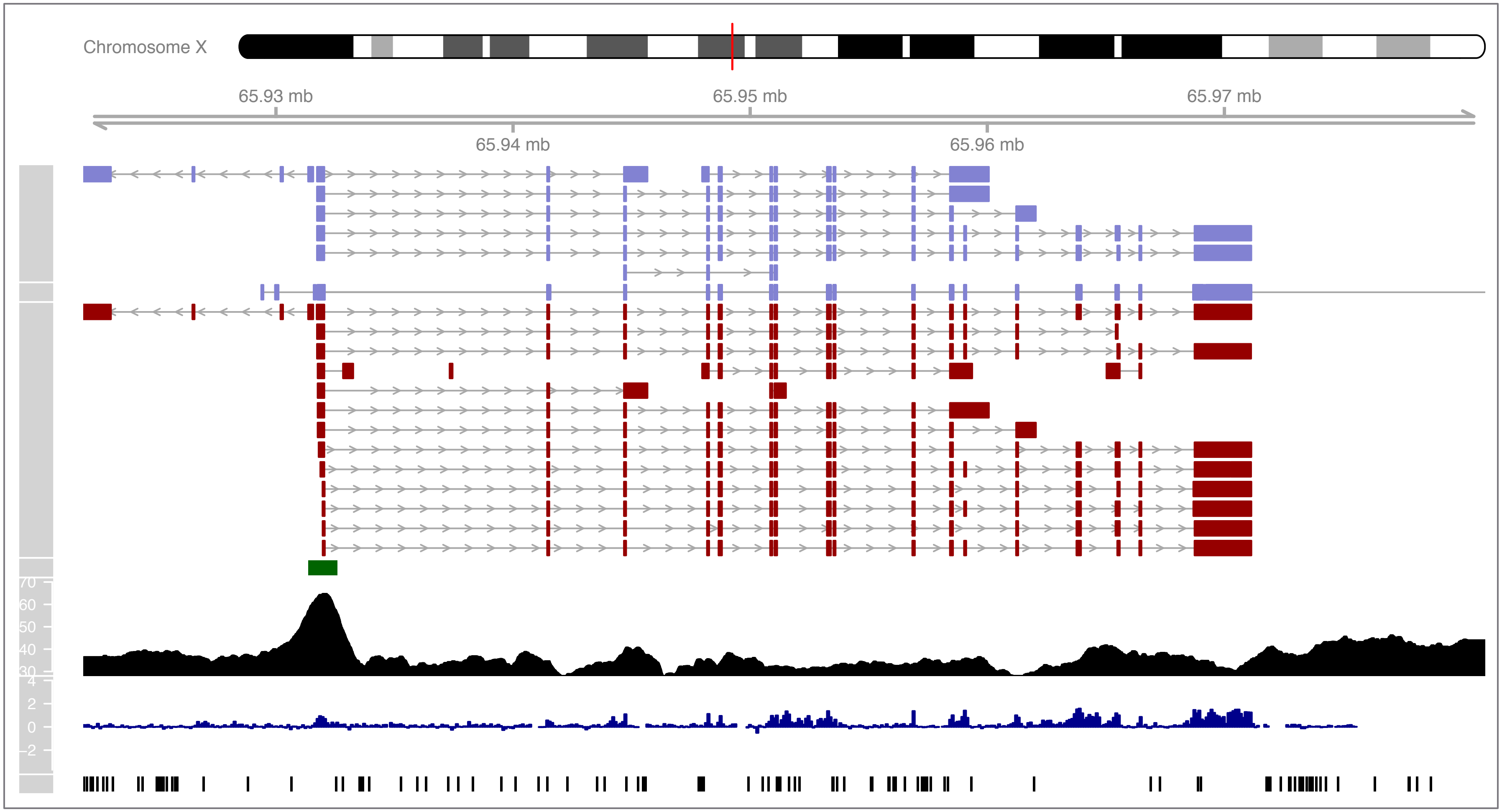
```
> gcContent <- UcscTrack(genome="mm9", chromosome="chrX",
+     track="GC Percent", table="gc5Base", from=from, to=to,
+     trackType="DataTrack", start="start", end="end", data="score",
+     type="hist", window=-1, windowSize=1500, fill.histogram="black",
+     col.histogram="black", ylim=c(30, 70), name="GC Percent")
```

Other tracks and plotting:

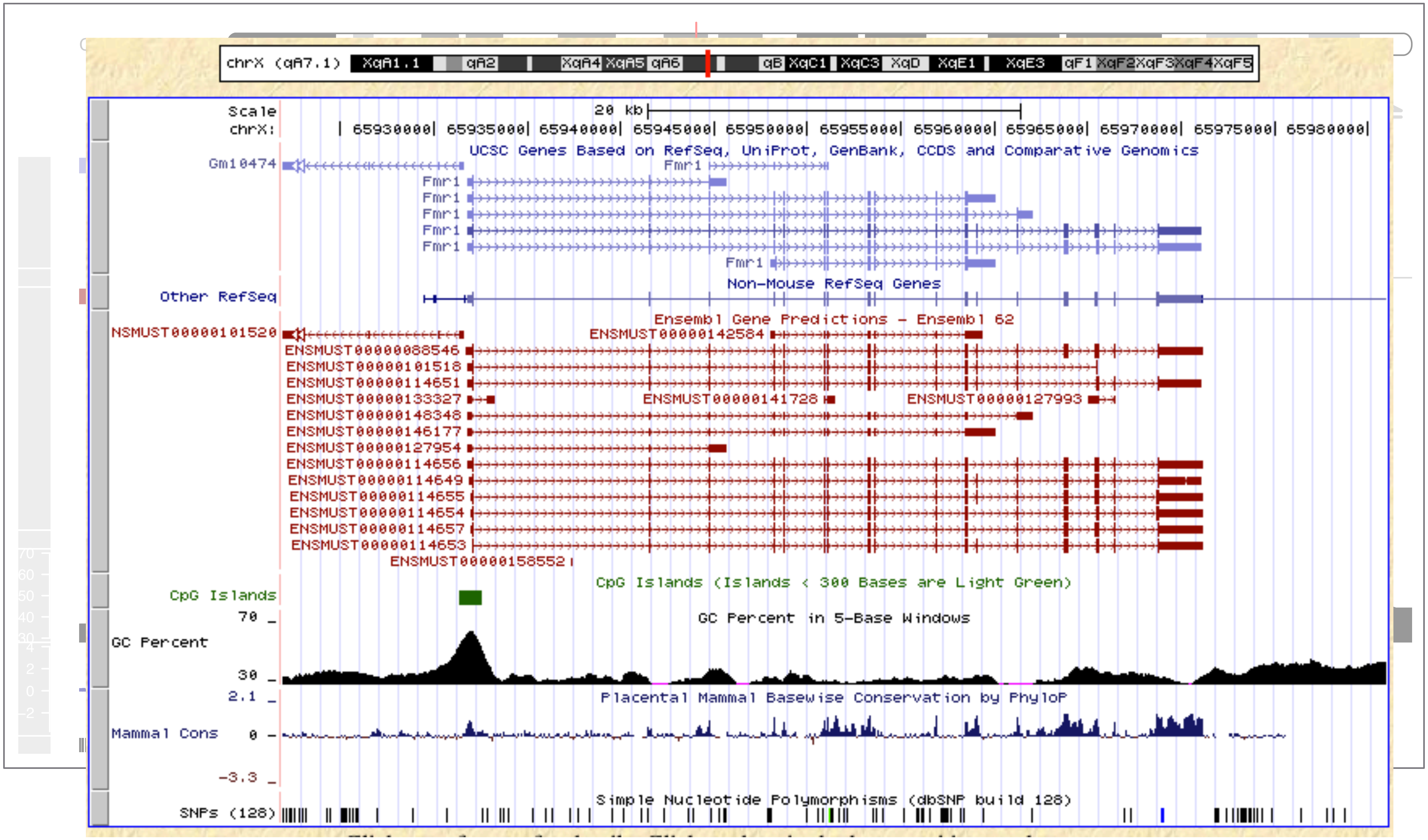
```
> axTrack <- GenomeAxisTrack()
> idxTrack <- IdeogramTrack(genome="mm9", chromosome="chrX")

> plotTracks(list(idxTrack, axTrack, knownGenes, refGenes,
+     ensGenes, cpgIslands, gcContent, conservation,
+     snpLocations), from=from, to=to, showTitle=FALSE)
```

Fetching data directly from UCSC



Fetching data directly from UCSC



Acknowledgements

- James Bullard and Steffen Durinck for the original software
- Arne Müller, Robert Ivanek, Steve Lianoglou for code contributions and discussions
- Martin Morgan, Nishant Gopalkrishnan, Marc Carlson and the rest of the Bioconductor crew for their great software suite and quickly implementing the missing bits and pieces
- Michael Lawrence and Tengfei Yin for their feedback
- PCS Informatics and the PCS management