

# Package ‘Pviz’

April 16, 2019

**Type** Package

**Title** Peptide Annotation and Data Visualization using Gviz

**Version** 1.16.0

**Author** Renan Sauteraud, Mike Jiang, Raphael Gottardo

**Maintainer** Renan Sauteraud <rsautera@fhcrc.org>

**Description** Pviz adapts the Gviz package for protein sequences and data.

**License** Artistic-2.0

**Depends** R(>= 3.0.0), Gviz(>= 1.7.10)

**Imports** biovizBase, Biostrings, GenomicRanges, IRanges, data.table,  
methods

**Suggests** knitr, pepDat

**biocViews** Visualization, Proteomics, Microarray

**VignetteBuilder** knitr

**git\_url** <https://git.bioconductor.org/packages/Pviz>

**git\_branch** RELEASE\_3\_8

**git\_last\_commit** c6a6fbc

**git\_last\_commit\_date** 2018-10-30

**Date/Publication** 2019-04-15

## R topics documented:

ATrack . . . . .	2
CladeTrack . . . . .	2
DTrack . . . . .	3
plot_clade . . . . .	4
plot_inter . . . . .	5
ProbeTrack . . . . .	5
ProteinAxisTrack . . . . .	7
ProteinSequenceTrack . . . . .	7

<b>Index</b>	<b>9</b>
--------------	----------

---

 ATrack

*ATrack class*


---

### Description

This class contains Gviz's AnnotationTrack and adds default values to the genome and chromosome slot

### Usage

```
ATrack(range = NULL, start = NULL, end = NULL, width = NULL, group, id,
        stacking = "squish", name = "ATrack", fun, selectFun, ...)
```

### Arguments

range, start, end, width, group, id, stacking, name, fun, selectFun, ...  
Arguments to be passed to AnnotationTrack.

### Author(s)

Renan Sauteraud

### See Also

[AnnotationTrack](#), [GdObject](#)

### Examples

```
# Object construction
aTrack <- ATrack(start = c(20, 60), end = c(40, 100), name = "random.anno",
                id=c("small","big"))
#Stacking example
a2Track <- ATrack(start = c(20, 30), end = c(40, 100), name = "stacking=dense",
                id = c("small", "big"), stacking = "dense", fill=c("black", "orange"))
a3Track <- ATrack(start = c(20, 30), end = c(40, 100), name = "no stacking",
                id = c("small", "big"), fill = c("black", "orange"))
#Plotting
plotTracks(trackList = c(aTrack, a2Track, a3Track), showFeatureId = TRUE)
```

---

 CladeTrack

*CladeTrack*


---

### Description

This track can be used to display the result of pepStat analysis for a single clade. It contains DTrack.

### Usage

```
CladeTrack(restab, clade, name = clade, ...)
```

**Arguments**

restab	A data.frame. The result of a peptide microarray analysis, as returned by pepStat's restab function.
clade	A character. The clade to plot.
name	A character. The name of the track, used in the title panel when plotting. By default, the clade name.
...	Additional argument to be passed to DataTrack. They will be treated as display parameters.

**Slots**

clade A character. The clade to display.

**Author(s)**

Renan Sauteraud

**See Also**

DTrack

**Examples**

```
if(require(pepDat)){
  data(restab)
  ct <- CladeTrack(restab, clade = "M", type = "l", legend = TRUE)
  plotTracks(ct)
}
```

---

DTrack

*DTrack class*


---

**Description**

This class contains Gviz's DataTrack and adds default values to the genome and chromosome slot

**Usage**

```
DTrack(range = NULL, start = NULL, end = NULL, width = NULL, data,
        name = "DTrack", ...)
```

**Arguments**

range, start, end, width, data, name, ...  
Arguments to be passed to DataTrack.

**Details**

Refer to DataTrack for details regarding the constructor.

**Author(s)**

Renan Sauteraud

**See Also**[DataTrack](#), [GdObject](#)**Examples**

```
dTrack <- DTrack(start=seq(1,1000, len=100), width=10, data=matrix(runif(400),
  nrow=4), name="random data")
```

---

plot\_clade

*Plot frequency of response for a single clade.*


---

**Description**

Plot an axis and the frequency of response of a single selected clade.

**Usage**

```
plot_clade(restab, clade, sequence = NULL, from = 0,
  to = max(restab$position), ...)
```

**Arguments**

restab	A <code>data.frame</code> . The result of a peptide microarray analysis, as returned by <code>pepStat</code> 's <code>restab</code> function.
clade	A character. The clade to plot.
sequence	An optional character or <code>AAString</code> . The sequence of the <code>ProteinSequenceTrack</code> to plot. It should be the sequence of the reference genome used in the <code>peptideSet</code> that generated the <code>restab</code> .
from	A numeric, the start coordinate of the plot.
to	A numeric, the end coordinate of the plot.
...	Additional arguments to be passed to <code>plotTracks</code> .

**Author(s)**

Renan Sauteraud

**See Also**`restab`, `plot_inter`, [plotTracks](#)**Examples**

```
if(require(pepDat)){
  data(restab)
  plot_clade(restab, clade = c("A", "M"))
}
```

---

plot_inter	<i>Plot frequency of response for each group</i>
------------	--------------------------------------------------

---

**Description**

Plot an axis and the frequency of response of each group, averaged by peptides at each position.

**Usage**

```
plot_inter(restab, sequence = NULL, from = 0, to = max(restab$position),
  ...)
```

**Arguments**

restab	A <code>data.frame</code> . The result of a peptide microarray analysis, as returned by <code>pepStat</code> 's <code>restab</code> function.
sequence	A character or an <code>AAStrng</code> . If not <code>NULL</code> , the sequence of the <code>ProteinSequenceTrack</code> to plot. It should be the sequence of the reference genome used in the <code>peptideSet</code> that generated the <code>restab</code> .
from	A numeric, the start coordinate of the plot.
to	A numeric, the end coordinate of the plot.
...	Additional arguments to be passed to <code>plotTracks</code> .

**Author(s)**

Renan Sauteraud

**See Also**

`restab`, `plot_clade`, [plotTracks](#)

**Examples**

```
if(require(pepDat)){
  data(restab_aggregate)
  plot_inter(restab_aggregate)
}
```

---

ProbeTrack	<i>ProbeTrack</i>
------------	-------------------

---

**Description**

This track can be used to display the frequency of antibody binding for each probe on an array as predicted by `pepStat`'s function `makeCalls`.

**Usage**

```
ProbeTrack(sequence, intensity, probeStart, restab = NULL, group = NULL,
  name = "ProbeTrack", ...)
```

**Arguments**

sequence	A character vector. The sequence of peptides to display.
intensity	A numeric vector. The frequency of binding or the baseline corrected intensity for the peptides.
probeStart	A numeric vector. The start position of the peptides.
name	A character. The name of the track used in the title panel when plotting
restab	A data.frame containing all the above parameters, as outputted by pepStat's restab function.
group	A character. The group to display on the ProbeTrak. This is only required when restab is not NULL. See details section for more information.
...	Arguments to be passed to DataTrack.

**Details**

The vectors for the arguments `sequence`, `freq` and `probeStart` should be of the same length. If `restab` is provided, the three previous arguments will be ignored and `group` must be specified. `group` must be a valid column name in `restab`, data.frame.

**Slots**

sequence	A character vector. The probes sequence.
probeStart	A numeric vector. The start position of the probes.
intensity	A numeric vector. The frequency of response of each probe. Or the baseline corrected intensity of the signal.

**Author(s)**

Renan Sauteraud

**See Also**

[GdObject](#)

`restab`

**Examples**

```
if(require(pepDat)){
  data(restab)
  pt <- ProbeTrack(sequence = restab$peptide,
                   intensity = restab$group2,
                   probeStart = restab$start)
  plotTracks(pt)
  plotTracks(pt, from = 460, to = 560, legend=TRUE)
}
```

---

ProteinAxisTrack      *ProteinAxisTrack*

---

**Description**

A track to display an axis for protein or peptide sequences

**Usage**

```
ProteinAxisTrack(range = NULL, name = "Axis", addNC = FALSE, id = NULL,
...)
```

**Arguments**

```
range,name,id,...
                Arguments to be passed to GenomeAxisTrack.
addNC          A logical. If TRUE, display the Amino-terminal and Carboxyl-terminal ends
                on the axis.
```

**Author(s)**

Renan Sauteraud

**See Also**

[GenomeAxisTrack](#)

**Examples**

```
# Object construction
paxTrack <- ProteinAxisTrack()
pax2 <- ProteinAxisTrack(addNC=TRUE)
pax3 <- ProteinAxisTrack(littleTicks=TRUE)
# Plotting
plotTracks(c(paxTrack,pax2,pax3), from=1, to=100)
```

---

ProteinSequenceTrack      *ProteinSequenceTrack*

---

**Description**

A track to display peptides and protein sequences.

**Usage**

```
ProteinSequenceTrack(sequence = NULL, name = "Sequence", ...)
```

**Arguments**

```
sequence      A character or AAString of length one. The sequence to display.
name          A character. The name of the track used in the title panel when plotting
...           Additional items which will all be interpreted as display parameters.
```

**Author(s)**

Renan Sauteraud

**See Also**

[SequenceTrack](#), [DisplayPars](#)

**Examples**

```
if(require(pepDat)){
  data(pep_hxb2)
  hxb2_seq <- metadata(pep_hxb2)$sequence
  st<-ProteinSequenceTrack(sequence=hxb2_seq, name="env")

  # Plotting amino acids
  plotTracks(st, to = 20)

  # When the range becomes wider, only coloured squares are displayed
  plotTracks(st, to = 100)

  # When overplotting, a single line will mark the ProteinSequenceTrack
  plotTracks(st)
}
```



# Index

AnnotationTrack, [2](#)  
ATrack, [2](#)  
ATrack-class (ATrack), [2](#)

CladeTrack, [2](#)  
CladeTrack-class (CladeTrack), [2](#)

DataTrack, [4](#)  
DisplayPars, [8](#)  
DTrack, [3](#)  
DTrack-class (DTrack), [3](#)

GdObject, [2](#), [4](#), [6](#)  
GenomeAxisTrack, [7](#)

plot\_clade, [4](#)  
plot\_inter, [5](#)  
plotTracks, [4](#), [5](#)  
ProbeTrack, [5](#)  
ProbeTrack-class (ProbeTrack), [5](#)  
ProteinAxisTrack, [7](#)  
ProteinAxisTrack-class  
(ProteinAxisTrack), [7](#)  
ProteinSequenceTrack, [7](#)  
ProteinSequenceTrack-class  
(ProteinSequenceTrack), [7](#)

SequenceTrack, [8](#)