

# Package ‘gTrack’

July 22, 2015

**Title** Plotting multiple tracks of complex genomic data across multiple genomic windows.

**Version** 0.0.0.9000

**Maintainer** Marcin Imielinski <mimieliński@nygenome.org>

**Description** Object for plotting GRanges, RleList, UCSC file formats, and ffTrack objects in multi-track panels.

**Depends** R (>= 3.1.0),  
GenomicRanges (>= 1.8),  
gUtils

**License** GNU GPLv3

**LazyData** true

## R topics documented:

c . . . . .	2
clear . . . . .	2
colormap . . . . .	2
dat . . . . .	3
edgs . . . . .	3
edgs<- . . . . .	4
formatting . . . . .	4
formatting<- . . . . .	5
gTrack-class . . . . .	5
karyogram . . . . .	9
lengths . . . . .	10
mdata . . . . .	10
plot . . . . .	11
seqinfo . . . . .	12
seqinfo<- . . . . .	13
show,gTrack-method . . . . .	13
track.gencode . . . . .	13
track.splice . . . . .	14
vars . . . . .	15
[ . . . . .	15
<b>Index</b>	<b>16</b>

---

c

---

*c*


---

**Description**

Concatenate gTrack objects gt1, gt2, gt3 usage: `c(gt1, gt2, gt3)` # returns a gTrack object with the component tracks "stacked"

**Usage**

```
## S4 method for signature 'gTrack'
c(x, ..., recursive = FALSE)
```

**Author(s)**

Marcin Imielinski

---

clear

---

*clear*


---

**Description**

Clear data from gTrack object (makes into an empty container with the same seqinfo and display features) usage: `clear(gTrack)`

**Usage**

```
## S4 method for signature 'gTrack'
clear(.Object)
```

**Author(s)**

Marcin Imielinski

---

colormap

---

*colormap*


---

**Description**

Access colormap of gTrack object, this is a named list of named character vectors that specifies the field of the underlying GRanges object that will be used to map a set of values to a set of colors. usage: `colormap(gt)`

Set colormap of gTrack object, this is a named list of named character vectors that specifies the field of the underlying GRanges object that will be used to map a set of values to a set of colors.

usage: `colormap(gt)[1] = list(tumortype = c(lung = 'red', pancreatic = 'blue', colon = 'purple'))`

**Usage**

```
## S4 method for signature 'gTrack'  
colormap(.Object)  
  
## S4 replacement method for signature 'gTrack'  
colormap(.Object) <- value
```

**Author(s)**

Marcin Imielinski  
Marcin Imielinski

---

dat	<i>dat</i>
-----	------------

---

**Description**

Extract list of data contained inside a gTrack object gt. Each list item will contain either a GRanges, RleList, or path to a file.  
usage dat(gt)

**Usage**

```
## S4 method for signature 'gTrack'  
dat(.Object)
```

**Author(s)**

Marcin Imielinski

---

edgs	<i>edgs</i>
------	-------------

---

**Description**

Get edges list associated with gTrack object, is is a list of data.frames each which contains a data.frame specifying formatted edges, as pairs of indices into the underlying GRanges object \$from, \$to and additional formats \$col \$lwd \$lty  
usage: edgs(gt)

**Usage**

```
## S4 method for signature 'gTrack'  
edgs(.Object)
```

**Author(s)**

Marcin Imielinski

---

<code>edgs&lt;-</code>	<i><code>edgs&lt;-</code></i>
------------------------	-------------------------------

---

### Description

Set edges data.frame associated with associated with gTrack object, the data.frame that is being used to replace must have fields \$from, \$to, and can have optional fields \$lwd \$lty, \$col specifyign color and line type.

usage: `edgs(gt)[[1]] <- new.edges.`

### Usage

```
## S4 replacement method for signature 'gTrack'
edgs(.Object) <- value
```

### Author(s)

Marcin Imielinski

---

<code>formatting</code>	<i><code>formatting</code></i>
-------------------------	--------------------------------

---

### Description

Get data frame specifying formatting of gTrack object gt usage: `formatting(gt)` gt # will just display the formatting data.frame

If you want to access particular fields of the formatting data.frame, just use the "\$" accessor like you would for a data.frame

### Usage

```
## S4 method for signature 'gTrack'
formatting(.Object)
```

### Author(s)

Marcin Imielinski

---

formatting<-	<i>formatting&lt;-</i>
--------------	------------------------

---

**Description**

Set formatting of gTrack object gt

usage: #gt is a gTrack object formatting(td)\$height <- 2

#can also just use \$ directly, like for a data frame, which is more convenient td\$height <- 2

**Usage**

```
## S4 replacement method for signature 'gTrack'
formatting(.Object) <- value
```

**Author(s)**

Marcin Imielinski

---

gTrack-class	<i>S4 class for gTrack</i>
--------------	----------------------------

---

**Description**

Class gTrack defines a subtable object that wraps formatting information around genomic data (GRanges, GRangesList, RleLists, UCSC formats, ffTrack) and can be displayed in a "genome browser" style plot.

Arguments described as "formatting" are vectors. They are replicated (if necessary) to match the length of the object.

**Usage**

```
gTrack(...)
```

**Arguments**

data	This is an instance of one of the following objects: (1) GRanges (2) GRangesList (3) ffTrack or (4) character representing Bed, Wig, BigWig, or .rds GRanges file. It can also be a list of the above (if specifying multiplying tracks).
y.field	vector or scalar character specifying meta data field of GRanges or GRangesList to use for "y axis" coordinate when plotting numeric tracks, (note: for a RleList or ffTrack this field is automatically set to "score"), default is NA for non-numeric tracks
name	vector or scalar character specifying name of this track, which will be displayed on label to the left of the track
height	vector or scalar numeric specifying height of track(s) (in relative units)
gr.labelfield	vector or scalar character specifying which GRanges meta data field to use for GRanges label (default "label") (formatting)

<code>gr1.labelfield</code>	vector or scalar character specifying which GRanges meta data field to use for GRangesList label (default "label") (formatting)
<code>legend</code>	vector or scalar logical specifying whether to draw a legend for this track (formatting)
<code>legend.xpos</code>	vector or scalar numeric between 0 and 1 specifying what relative x position in the plot to place the legend for this track (formatting)
<code>legend.ypos</code>	vector or scalar numeric between 0 and 1 specifying what relative y position in the plot to place the legend for this track (formatting)
<code>legend.ncol</code>	vector or scalar positive integer specifying how many columns to put in legend (formatting)
<code>legend.xjust</code>	vector or scalar of 0, 1, 2 specifying x justification of legend (formatting)
<code>legend.yjust</code>	vector or scalar of 0, 1, 2 specifying y justification of legend (formatting)
<code>legend.maxitems</code>	vector or scalar positive integer specifying what is the maximum number of items to include in legend (formatting)
<code>label.suppress</code>	vector or scalar logical flag specifying whether to suppress all GRanges / GRangesList label drawing (formatting)
<code>label.suppress.gr</code>	vector or scalar logical flag specifying whether to suppress GRanges label drawing (formatting)
<code>label.suppress.gr1</code>	vector or scalar logical flag specifying whether to suppress GRangesList label drawing (formatting)
<code>ygap</code>	vector or scalar numeric specifying gap between tracks
<code>stack.gap</code>	vector or scalar numeric specifying x gap between stacking non-numeric GRanges or GRangesLists items in track(s)
<code>ywid</code>	vector or scalar numeric specifying the y-extent of individual ranges (in local plot coordinates)
<code>col</code>	vector or scalar character specifying static color for track(s), if NA then color is specified by <code>colormaps()</code> property or <code>gr.colorfield</code> or <code>col</code> meta data field of GRanges / GRangesList data object
<code>border</code>	vector or scalar character specifying static border for polygons in track(s), if NA then <code>\$border</code> is determined using <code>gr.colorfield</code> / <code>colormap</code> or meta field <code>\$border</code> of GRanges / GRangesList
<code>max.ranges</code>	vector or scalar numeric specifying what is the max number of ranges to plot in a window (formatting)
<code>angle</code>	vector or scalar numeric specifying angle of polygons that represent signed range'
<code>lift</code>	vector or scalar logical flag specifying whether to lift this track to other chained-Track items (only relevant if used within <code>chainedTracks</code> object)
<code>split</code>	vector or scalar logical flag specifying whether to split when lifting (only relevant if used within <code>chainedTracks</code> object)
<code>colormaps</code>	<code>length(.Object)</code> length named list of named vectors whose entry <code>i</code> maps unique value of a data field to colors. The <code>data.field</code> is specified by the name of the list entry, and the unique values / colors are specified by the named vector.

edges	Data frame of columns \$from, \$to, and optional fields \$col, \$lwd, and \$lty, specifying edges linking data items. Also can be a list of the above if specifying multiple tracks (and must be compatible in length with data arg)
xaxis.suffix	vector or scalar numeric specifying the suffix that will be used in describing x axis coordinates (TODO: move to display) (formatting)
xaxis.unit	vector or scalar numeric specifying the unit that will be used in describing x axis coordinates (TODO: move to display) (formatting)
xaxis.round	vector or scalar non-neg integer specifying number of decimals to round xaxis coordinate labels (formatting)
xaxis.nticks	vector or scalar positive integer specifying how many xaxis ticks to optimally draw (formatting)
xaxis.label.angle	vector or scalar numeric between 0 and 360 specifying angle with which to draw xaxis coordinate labels (formatting)
xaxis.newline	vector or scalar logical specifying whether to draw a newline in the xaxis coordinate labels (formatting)
lwd.border	vector or scalar integer specifying the thickness of the polygon borders (formatting)
cex.label	vector or scalar numeric specifying the expansion factor of the range labels (formatting)
hadj.label	vector or scalar numeric specifying the horizontal adjustment of the range labels (formatting)
vadj.label	vector or scalar numeric specifying the vertical adjustment of the range labels (formatting)
ypad	vector or scalar numeric between 0 and 1 specifying how much whitespace padding to add within panel (formatting)
circles	vector or scalar logical specifying whether to scatter plot range data (formatting)
lines	vector or scalar logical specifying whether to line plot range data (formatting)
bars	vector or scalar logical specifying whether to bar plot range data (formatting)
y0.bar	vector or scalar numeric specifying where to draw the lower boundary of a bar in a bar plot (only applicable if bars == T) (formatting)
source.file.chrsub	vector or scalar logical specifying whether or not sub "chr" out of any external files (e.g. UCSC style files) (formatting)
y.grid.col	vector or scalar character specifying color of "gridlines" used to specify numeric track data (formatting)
y.grid.cex	vector or scalar non-neg numeric specifying character expansion for y tick / y grid labels (formatting)
y.grid.lty	vector or scalar positive integer specifying line style of y grid lines for numeric tracks (formatting)
y.grid.lwd	vector or scalar positive integer specifying thickness of y grid lines for numeric tracks (formatting)
y.grid.labx	vector or scalar positive integer specifying fraction of xlim left of plot to place y axis labels (formatting)
yaxis	vector or scalar logical specifying whether to print yaxis (formatting)

<code>yaxis.pretty</code>	vector or scalar positive integer specifying how many ticks to optimally draw on yaxis (formatting)
<code>draw.var</code>	vector or scalar logical specifying whether to draw.var for GRanges / GRanges-List specifying reads (GRanges must contain \$cigar +/- \$MD field) (formatting)
<code>draw.paths</code>	vector or scalar logical specifying whether to interpret GRangesLists as "paths" and connect them with a set of spline curves. (formatting)
<code>path.col</code>	vector or scalar character specifying color of path (only applicable for tracks in which draw.paths = T) (formatting)
<code>path.col.arrow</code>	vector or scalar character specifying color of arrow of path (only applicable for tracks in which draw.paths = T) (formatting)
<code>path.cex.arrow</code>	vector or scalar numeric > 0 specifying expansion factor of arrow of path (only applicable for tracks in which draw.paths = T) (formatting)
<code>path.stack.y.gap</code>	vector or scalar numeric > 0 specifying y stack gap of paths (only applicable for tracks in which draw.paths = T) (formatting)
<code>path.stack.x.gap</code>	vector or scalar numeric > 0 specifying x stack gap for paths (only applicable for tracks in which draw.paths = T) (formatting)
<code>path.cex.v</code>	vector or scalar numeric > 0 specifying vertical bulge of spline in paths (only applicable for tracks in which draw.paths = T) (formatting)
<code>path.cex.h</code>	vector or scalar numeric > 0 specifying horizontal bulge of spline in paths (formatting) (only applicable for tracks in which draw.paths = T) (formatting)
<code>draw.backbone</code>	vector or scalar logical specifying whether to draw "backbone" connecting different items in a GRangesList item (formatting)
<code>cex.tick</code>	vector or scalar numeric specifying expansion factor for axis tick labels (formatting)
<code>cex.tick.len</code>	vector or scalar numeric specifying lengths for axis ticks (formatting)
<code>gr.cex.label</code>	vector or scalar numeric specifying GRanges label character expansion (default is cex.label) (formatting)
<code>gr.srt.label</code>	vector or scalar numeric between 0 and 180 specifying rotation of GRanges labels (formatting)
<code>sep.lty</code>	vector or scalar integer specifying line style for window separators (formatting)
<code>sep.lwd</code>	vector or scalar numeric specifying line thickness for window separators (formatting)
<code>cmap.min</code>	minimum saturating data value of color map for triangle plot
<code>cmap.max</code>	maximum saturating data value of color map for triangle plot

## Slots

**data** `length(.Object)` length list containing genomic data (e.g. GRanges, GrangesLists, RleLists, path to ucsc file or ffTrack file on disk)

**seqinfo** Seqinfo object

**colormap** `length(.Object)` length named list of named vectors whose entry i specifies the colormap for the meta data field of object entry i (specified by the name) and maps unique value of that data field to colors (specified by the named vector)

**edges** list of data.frames of length `length(.Object)` which has columns \$from, \$to, and optional fields \$col, \$lwd, and \$lty to specify splined edges joining data items in the corresponding track



**Author(s)**

Marcin Imielinski

Marcin Imielinski

---

karyogram

---

*karyogram***Description**

Returns gTrack displaying band pattern for hg18 or hg19 karyotype

Returns gTrack object representing refGene transcripts and their components (utr, cds etc) with assigned colors. Usually built from cahced data objects but can also be built from provided GRanges-List

**Usage**

```
karyogram(hg19 = T, bands = T, arms = T, tel.width = 2e+06, ...)
```

```
track.refgene(rg = NULL, gene.collapse = T, genes = NULL,
  bg.col = alpha("blue", 0.1), cds.col = alpha("blue", 0.6),
  utr.col = alpha("purple", 0.4), st.col = "green", en.col = "red",
  genespan = T, utr = T, cds = T, grl.labelfield, gr.labelfield, col,
  cached = T, cached.path = system.file("extdata", "refgene.composite.rds",
  package = "gTrack"), cached.path.collapsed = system.file("extdata",
  "refgene.composite.collapsed.rds", package = "gTrack"), gr.srt.label = 0,
  gr.cex.label = 0.8, labels.suppress.gr = T, stack.gap = 1e+06, ...)
```

**Arguments**

hg19	logical scalar flag, if T returns gTrack for hg19
bands	logical scalar, if T returns gTrack with colored giemsa bands
arms	logical scalar, if T and bands F, returns chromosome arms with different colors and centromeres and telomeres marked, if arms is F and bands F returns chromosomes, each with a different color
tel.width	numeric scalar, specifies telomere width in bases (only relevant if arms = T, bands = F)
...	additional arguments passed down to gTrack
rg	(optional) GRangesList representing transcript models obtained from refgene, with GrangesList meta data fields chr, s1, s2, e1, e2, str, gene_sym, Uniprot,
gene.collapse	scalar logical specifying whether to collapse genes by transcript (or used stored version of transcripts)
genes	(optional) character vector specifying genes to limit gTrack object to
bg.col	scalar character representing background color for genespan
cds.col	scalar character representing background color for CDS
st.col	scalar character representing color of CDS start
en.col	scalar character representing color of CDS end

genespan	logical scalar whether to include genespan range around entire gne
utr	logical scalar whether to include range specifying UTR
cds	logical scalar whether to include range specifying CDS
cached	logical scalar whether to use "cached" version provided with package
gr.srt.label	scalar numeric specifying angle on exon label
gr.cex.label	scalar numeric > 0 specifying character expansion on exon label
labels.suppress.gr	scalar logical specifying whether to suppress exon label plotting
stack.gap	stack.gap argument to gTrack
cds.utr	scalar character representing background color for UTR

**Author(s)**

Marcin Imielinski

Marcin Imielinski

---

lengths	<i>lengths</i>
---------	----------------

---

**Description**

gets lengths of data objects inside gTrack object gt usage: lengths(gt) # returns vector of lengths

**Usage**

```
## S4 method for signature 'gTrack'
lengths(x)
```

**Author(s)**

Marcin Imielinski

---

mdata	<i>mdata</i>
-------	--------------

---

**Description**

Accessing submatrices of mdata object for length 1 gTrack gt

Usage: (igr, jgr are GRanges objects corresponding to slices of matrix to be accessed from gt)  
mdata(gt, igr, jgr)

Accessing columns of gTrack formatting data.frame

Setting formats of gTrack object - ie modifying the formatting(gt) data frame after an object has already been instantiated

Usage: gt\$y.field = 'score' gt\$gr.colorfield[1] = 'readtype'

Getting length of gTrack object gt Usage: length(gt)

Computing the GRanges footprint of the gTrack object on the genome usage: reduce(gt) # outputs a GRanges

**Usage**

```
## S4 method for signature 'gTrack'
mdata(x, igr = NULL, jgr = igr)

## S4 method for signature 'gTrack'
x$name

## S4 replacement method for signature 'gTrack'
x$name <- value

## S4 method for signature 'gTrack'
length(x)

## S4 method for signature 'gTrack'
reduce(x, ...)
```

**Arguments**

... additional arguments to GRanges reduce function

**Author(s)**

Jeremiah Wala  
 Marcin Imielinski  
 Marcin Imielinski  
 Marcin Imielinski  
 Marcin Imielinski

---

plot

---

plot

---

**Description**

Plot gTrack object in multi-track genome browser view. gTracks will appear as stacked interval plots, line / bar / scatter plots, node-edge graphs, or triangle plots depending on the formatting settings and additional data (eg mdata) provided. gTracks can be drawn across several non-contiguous "windows" of the genome

Additional argument "links" takes a GRangesList of signed interval pairs (ie each item is length 2) representing genomic junctions as input. The junctions will be pointing right for + intervals and left for - intervals. The meta data of the links GRangesList (\$lwd \$col \$lty) can be set to modulate the color, width, and line style of the junction links .

usage: display(gt, win) # where win is a GRanges object display(gt) # this will show the entire span of seqinfo(gt) display(gt, '3:1e6-2e6') ## can use UCSC style strings display(gt, GRanges(20, IRanges(20e6, 21e6))) display(gt, '2', links = ra) # here, the entire chromosome 2 is shown, with rearrangements on top

**Usage**

```
## S4 method for signature 'gTrack,ANY'
plot(x, y = seqinfo2gr(seinfo(x)), links = NULL,
     gap = NULL, y.heights = NULL, y.gaps = NULL, cex.xlabel = 1,
     cex.ylabel = 1, max.ranges = NA, links.feats = NULL, windows = NULL,
     verbose = FALSE, ...)
```

**Arguments**

<code>links</code>	GRangesList of signed locus pairs specifying links to draw above the plot, optional GRangesList values meta data specify formatting of individual links: <code>\$label</code> (text label), <code>\$col</code> (line color), <code>\$lwd</code> (line weight), <code>\$lty</code> (line style), <code>\$arrow</code> (arrow flag), <code>\$col.arrow</code> (arrow color), <code>\$v</code> (vertical spline bulge), <code>\$w</code> (horizontal spline bulge)
<code>gap</code>	scalar numeric specifying gap between windows (only relevant if windows has <code>length &gt; 1</code> ). Units of the gap are in genome coordinates
<code>max.ranges</code>	scalar numeric $> 0$ specifying max number of ranges to draw in a window (via sampling). If specified, overrides gTrack <code>max.ranges</code> formatting feature.
<code>windows</code>	GRanges specifying windows to view (can also be GRangesList), default is whole genome
<code>...</code>	additional last-minute formatting changes to the gtrack can be entered here (eg <code>col = 'blue'</code> )
<code>cex.tick</code>	vector specifying relative heights of gaps below and above (stacked) windows, only relevant if windows is GRangesList

**Author(s)**

Marcin Imielinski, Jeremiah Wala

---

seqinfo

*seqinfo*


---

**Description**

returns Seqinfo of gTrack object `gt` Usage: `seqinfo(gt)`

**Usage**

```
## S4 method for signature 'gTrack'
seqinfo(x)
```

**Author(s)**

Marcin Imielinski

---

seqinfo<-	<i>seqinfo,-</i>
-----------	------------------

---

**Description**

set seqinfo property of gTrack

**Usage**

```
## S4 replacement method for signature 'gTrack'
seqinfo(.Object) <- value
```

**Author(s)**

Marcin Imielinski

---

show,gTrack-method

---

**Usage**

```
## S4 method for signature 'gTrack'
show(object)
```

**Author(s)**

Marcin Imielinski

---

track.gencode	<i>track.gencode</i>
---------------	----------------------

---

**Description**

Returns gTrack object representing GENCODE transcripts and their components (utr, cds etc) with assigned colors. Usually built from cached data objects but can also be built from provided GRangesList

**Usage**

```
track.gencode(gencode = NULL, gene.collapse = T, genes = NULL,
  grep = NULL, grepe = NULL, bg.col = alpha("blue", 0.1),
  cds.col = alpha("blue", 0.6), utr.col = alpha("purple", 0.4),
  st.col = "green", en.col = "red", gr1.labelfield, gr2.labelfield, col,
  cached = T, cached.path = system.file("extdata", "gencode.composite.rds",
  package = "gTrack"), cached.path.collapsed = system.file("extdata",
  "gencode.composite.collapsed.rds", package = "gTrack"), gr.srt.label = 0,
  gr.cex.label = 0.3, cex.label = 0.5, labels.suppress.gr = T,
  drop.rp11 = TRUE, stack.gap = 1e+06, ...)
```

**Arguments**

gene.collapse	scalar logical specifying whether to collapse genes by transcript (or used stored version of transcripts)
genes	(optional) character vector specifying genes to limit gTrack object to
grep	character vector for which to grep genes to positively select
grepe	character vector for which to grep genes which to exclude
bg.col	scalar character representing background color for genespan
cds.col	scalar character representing background color for CDS
st.col	scalar character representing color of CDS start
en.col	scalar character representing color of CDS end
cached	logical scalar whether to use "cached" version provided with package
gr.srt.label	scalar numeric specifying angle on exon label
gr.cex.label	scalar numeric > 0 specifying character expansion on exon label
labels.suppress.gr	scalar logical specifying whether to suppress exon label plotting
stack.gap	stack.gap argument to gTrack
...	additional arguments passed down to gTrack
rg	(optional) GRangesList representing transcript models imported from GENCODE gff3 file using rtracklayer import
cds.utr	scalar character representing background color for UTR

**Author(s)**

Marcin Imielinski

---

track.splice	<i>track.splice</i> Given set of exons and rna bam (eg from tophat) determines junction and exon read density and returns a gTrack object of splicing graph
--------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------

---

**Description**

track.splice

Given set of exons and rna bam (eg from tophat) determines junction and exon read density and returns a gTrack object of splicing graph

**Usage**

```
track.splice(ex = NULL, region = NULL, bam, verbose = TRUE,
             infer.exons = FALSE, min.reads = 0, min.exon.width = 10,
             max.exon.width = 1000)
```

**Arguments**

ex	GRanges of candidate exons
bam	path to indexed RNA seq bam
verbose	

**Author(s)**

Marcin Imielinski

---

vars	vars
------	------

---

**Description**

Get variants associated with track

**Usage**

```
## S4 method for signature 'gTrack'  
vars(.Object)
```

**Author(s)**

Marcin Imielinski

---

[	/
---	---

---

**Description**

Subsetting tracks of gTrack gt gt[1] gt[ix] # where ix is an integer vector

**Usage**

```
## S4 method for signature 'gTrack,ANY,ANY'  
x[i]
```

**Author(s)**

Marcin Imielinski

# Index

[, [15](#)

c, [2](#)

clear, [2](#)

colormap, [2](#)

dat, [3](#)

edgs, [3](#)

edgs<-, [4](#)

formatting, [4](#)

formatting<-, [5](#)

gTrack (gTrack-class), [5](#)

gTrack-class, [5](#)

karyogram, [9](#)

lengths, [10](#)

mdata, [10](#)

plot, [11](#)

seqinfo, [12](#)

seqinfo<-, [13](#)

show, gTrack-method, [13](#)

track.gencode, [13](#)

track.refgene (karyogram), [9](#)

track.splice, [14](#)

vars, [15](#)