

Package ‘HilbertCurve’

April 23, 2016

Type Package

Title Making 2D Hilbert Curve

Version 1.0.0

Date 2015-10-8

Author Zuguang Gu

Maintainer Zuguang Gu <z.gu@dkfz.de>

Depends R (>= 3.1.2), grid, IRanges, GenomicRanges

Imports methods, HilbertVis, png, grDevices

Suggests knitr, testthat (>= 0.3), circlize (>= 0.3.1), ComplexHeatmap (>= 1.3.0), markdown

VignetteBuilder knitr

Description Hilbert curve is a type of space-filling curves that fold one dimensional axis into a two dimensional space, but with still keep the locality. This package aims to provide a easy and flexible way to visualize data through Hilbert curve.

biocViews Software, Visualization, Sequencing, Coverage, GenomeAnnotation

URL <https://github.com/jokergoo/HilbertCurve>

License GPL (>= 2)

Repository Bioconductor

Date/Publication 2015-10-8 00:00:00

NeedsCompilation no

R topics documented:

hc_layer-HilbertCurve-method	2
hc_level-HilbertCurve-method	3
hc_normal_points-HilbertCurve-method	4
hc_png-HilbertCurve-method	5
hc_points-HilbertCurve-method	6

hc_rect-HilbertCurve-method	7
hc_segmented_points-HilbertCurve-method	9
hc_segments-HilbertCurve-method	10
hc_text-HilbertCurve-method	11
HilbertCurve	12
HilbertCurve-class	13
show-HilbertCurve-method	14
unzoom-HilbertCurve-method	15
zoom-HilbertCurve-method	16

Index 17

hc_layer-HilbertCurve-method

Add a new layer to the Hilbert curve

Description

Add a new layer to the Hilbert curve

Usage

```
## S4 method for signature 'HilbertCurve'
hc_layer(object, ir, x1 = NULL, x2 = NULL, col = "red",
         mean_mode = c("w0", "absolute", "weighted"), grid_line = 0)
```

Arguments

object	A HilbertCurve-class object.
ir	a IRanges object.
x1	if positions are not integers, they can be set by x1 and x2.
x2	if positions are not integers, they can be set by x1 and x2.
col	colors corresponding to intervals in ir (or combinations of x1 and x2).
mean_mode	when a segment in the curve overlaps with intervals in ir, how to calculate the mean values for this segment. See explanation in hc_points .
grid_line	whether add grid lines to show blocks of the Hilber curve. It should be an integer number and there will be $2^{(\text{grid_line}-1)}-1$ grid lines horizontal and vertical.

Details

If you want to add more than one layers to the curve, remember to set colors transparent.

This function only works under 'pixel' mode.

Value

No value is returned.

Author(s)

Zuguang Gu <z.gu@dkfz.de>

Examples

```
hc = HilbertCurve(1, 100, level = 9, mode = "pixel")
x = sort(sample(100, 20))
s = x[1:10*2 - 1]
e = x[1:10*2]
ir = IRanges(s, e)
hc_layer(hc, ir)
hc = HilbertCurve(1, 100, level = 9, mode = "pixel")
hc_layer(hc, ir, grid_line = 3)
```

hc_level-HilbertCurve-method

Level of the Hilbert curve

Description

Level of the Hilbert curve

Usage

```
## S4 method for signature 'HilbertCurve'
hc_level(object)
```

Arguments

object A [HilbertCurve-class](#) object.

Value

The level of the Hilbert curve.

Author(s)

Zuguang Gu <z.gu@dkfz.de>

Examples

```
hc = HilbertCurve(1, 100)
hc_level(hc)
hc = HilbertCurve(1, 100, level = 5)
hc_level(hc)
```

hc_normal_points-HilbertCurve-method

Add points to the Hilbert curve

Description

Add points to the Hilbert curve

Usage

```
## S4 method for signature 'HilbertCurve'  
hc_normal_points(object, ir, x1 = NULL, x2 = NULL, gp = gpar(),  
  pch = 1, size = unit(1, "char"))
```

Arguments

object	A HilbertCurve-class object.
ir	a IRanges object.
x1	if positions are not integers, they can be set by x1 and x2.
x2	if positions are not integers, they can be set by x1 and x2.
size	size of the points. It should be a unit object.
pch	shape of points.
gp	graphical parameters for points. It should be specified by gpar .

Details

Points are added at the middle of the intervals in `ir`.

This function is used internally.

Value

A data frame which contains coordinates for points.

Author(s)

Zuguang Gu <z.gu@dkfz.de>

Examples

```
# see documentation of hc_points  
NULL
```

hc_png-HilbertCurve-method

Save Hilbert curve as a PNG figure

Description

Save Hilbert curve as a PNG figure

Usage

```
## S4 method for signature 'HilbertCurve'  
hc_png(object, file = "Rplot.png")
```

Arguments

object	A HilbertCurve-class object.
file	file name. If the suffix of the file name is not .png, it will be added automatically no matter you like it or not.

Details

A PNG figure with resolution of $2^{\text{level}} \times 2^{\text{level}}$ is generated.

Only the body of the Hilbert curve will be written to PNG file.

This function only works under 'pixel' mode.

Value

No value is returned.

Author(s)

Zuguang Gu <z.gu@dkfz.de>

Examples

```
hc = HilbertCurve(1, 100, level = 9, mode = "pixel")  
x = sort(sample(100, 20))  
s = x[1:10*2 - 1]  
e = x[1:10*2]  
ir = IRanges(s, e)  
hc_layer(hc, ir)  
hc_png(hc, file = "test.png")
```

 hc_points-HilbertCurve-method

Add points to the Hilbert curve

Description

Add points to the Hilbert curve

Usage

```
## S4 method for signature 'HilbertCurve'
hc_points(object, ir, x1 = NULL, x2 = NULL,
  np = max(c(2, 10 - hc_level(object))), size = unit(1, "char"),
  pch = 1, gp = gpar(), mean_mode = c("w0", "absolute", "weighted"),
  shape = c("circle", "square", "triangle", "hexagon", "star"))
```

Arguments

object	A HilbertCurve-class object.
ir	a IRanges object.
x1	if positions are not integers, they can be set by x1 and x2.
x2	if positions are not integers, they can be set by x1 and x2.
np	number of points (a circle or a square, ...) that are put in a segment. np controls the mode of how to add the points to the curve. See 'details' section.
size	size of the points. It should be a unit object. Only works if np < 2
pch	shape of points, used for points if np < 2.
gp	graphical parameters for points. It should be specified by gpar .
mean_mode	when a segment in the curve overlaps with intervals in ir, how to calculate the mean values for this segment (such as the RGB colors). See 'Details' section for a detailed explanation.
shape	shape of points, used for points if np >= 2.

Details

If np is set to a value less than 2 or NULL, points will be added at the middle points in ir (or x1, x2). If np is set to a value larger or equal to 2, every segment that overlaps to ir will be segmented into np parts and a circle (or star, ...) is put on every 'small segments'.

Following illustrates different settings for mean_mode:

100	80	60	values in ir (e.g. red component for colors)
+++++	+++	+++++	ir
=====			window (width = 16)
4	3	3	overlap

```

absolute: (100 + 80 + 60)/3
weighted: (100*4 + 80*3 + 60*3)/(4 + 3 + 3)
w0:      (100*4 + 80*3 + 60*3)/16

```

So use of the mode depends on specific scenario. For example, if `ir` corresponds to positions of genes, then the mode of `w0` is perhaps a good choice. If `ir` corresponds to positions of CpG sites which is has width of 1 and most of the time is sparse in genomic windows, then `absolute` is a correct choice.

Graphical parameters can be set as a vector and they will be averaged according to above rules.

Internally, it will depatch to `hc_normal_points,HilbertCurve-method` or `hc_segmented_points,HilbertCurve-method` depending on the value of `np`.

Value

A data frame which contains coordinates for points.

Author(s)

Zuguang Gu <z.gu@dkfz.de>

Examples

```

hc = HilbertCurve(1, 100, level = 4, reference = TRUE)
x = sort(sample(100, 20))
s = x[1:10*2 - 1]
e = x[1:10*2]
ir = IRanges(s, e)
hc_points(hc, ir)
hc = HilbertCurve(1, 100, level = 4, reference = TRUE)
hc_points(hc, x1 = c(1.5, 50.5), x2 = c(10.5, 60.5))
require(circlize)
value = runif(length(ir))
col_fun = colorRamp2(range(value), c("white", "red"))
hc = HilbertCurve(1, 100, level = 4, reference = TRUE)
hc_points(hc, ir, np = 3, shape = "star", gp = gpar(fill = col_fun(value)))
hc = HilbertCurve(1, 100, level = 4, reference = TRUE)
hc_points(hc, ir, np = 0)
hc = HilbertCurve(1, 100, level = 4, reference = TRUE)
hc_points(hc, np = 0, x1 = c(1.5, 50.5), x2 = c(10.5, 60.5))
hc_points(hc, np = 0, x1 = 70.5, gp = gpar(col = "red"))

```

hc_rect-HilbertCurve-method

Add rectangles on Hilbert curve

Description

Add rectangles on Hilbert curve

Usage

```
## S4 method for signature 'HilbertCurve'  
hc_rect(object, ir, x1 = NULL, x2 = NULL,  
        gp = gpar(fill = "red", col = "red"),  
        mean_mode = c("w0", "absolute", "weighted"))
```

Arguments

object	A HilbertCurve-class object.
ir	a IRanges object.
x1	if positions are not integers, they can be set by x1 and x2.
x2	if positions are not integers, they can be set by x1 and x2.
gp	graphical parameters for rectangles. It should be specified by gpar .
mean_mode	when a segment in the curve overlaps with intervals in ir, how to calculate the mean values for this segment. See explanation in hc_points .

Details

You cannot set the width or height of the rectangles. Rectangles are always located at the turning points and have width or height equal to the length of the segments.

Value

A data frame which contains coordinates for rectangles.

Author(s)

Zuguang Gu <z.gu@dkfz.de>

Examples

```
hc = HilbertCurve(1, 100, level = 4, reference = TRUE)  
x = sort(sample(100, 20))  
s = x[1:10*2 - 1]  
e = x[1:10*2]  
ir = IRanges(s, e)  
hc_rect(hc, ir)
```

hc_segmented_points-HilbertCurve-method
Add points to the Hilbert curve

Description

Add points to the Hilbert curve

Usage

```
## S4 method for signature 'HilbertCurve'
hc_segmented_points(object, ir, x1 = NULL, x2 = NULL, gp = gpar(),
  np = max(c(2, 10 - hc_level(object))),
  mean_mode = c("w0", "absolute", "weighted"),
  shape = c("circle", "square", "triangle", "hexagon", "star"))
```

Arguments

object	A HilbertCurve-class object.
ir	a IRanges object.
x1	if positions are not integers, they can be set by x1 and x2.
x2	if positions are not integers, they can be set by x1 and x2.
np	number of points (a circle or a square, ...) that are put in a segment.
gp	graphical parameters for points. It should be specified by gpar .
mean_mode	when a segment in the curve overlaps with intervals in ir, how to calculate the mean values for this segment. See explanation in hc_points .
shape	shape of points. Possible values are "circle", "square", "triangle", "hexagon", "star".

Details

Every segment that overlaps to ir will be segmented into np parts and a circle (or star, ...) is put on every 'small segments'.

This function is used internally.

Value

A data frame which contains coordinates for points.

Author(s)

Zuguang Gu <z.gu@dkfz.de>

Examples

```
# see documentation of hc_points
NULL
```

hc_segments-HilbertCurve-method

Add line segments to Hilbert curve

Description

Add line segments to Hilbert curve

Usage

```
## S4 method for signature 'HilbertCurve'  
hc_segments(object, ir, x1 = NULL, x2 = NULL,  
            gp = gpar(lty = 1, lwd = 1, col = 1))
```

Arguments

object	A HilbertCurve-class object.
ir	a IRanges object.
x1	if positions are not integers, they can be set by x1 and x2.
x2	if positions are not integers, they can be set by x1 and x2.
gp	graphical parameters for lines. It should be specified by gpar .

Value

A data frame which contains coordinates for segments.

Author(s)

Zuguang Gu <z.gu@dkfz.de>

Examples

```
hc = HilbertCurve(1, 100, level = 4, reference = TRUE)  
x = sort(sample(100, 20))  
s = x[1:10*2 - 1]  
e = x[1:10*2]  
ir = IRanges(s, e)  
hc_segments(hc, ir)
```

hc_text-HilbertCurve-method

Add text to Hilbert curve

Description

Add text to Hilbert curve

Usage

```
## S4 method for signature 'HilbertCurve'  
hc_text(object, ir, labels, x1 = NULL, x2 = NULL, gp = gpar(), ...)
```

Arguments

object	A HilbertCurve-class object.
ir	a IRanges object that contains positions of text. Basically, the middle point of the interval will be the position of the text.
labels	text corresponding to intervals in <code>ir</code> .
x1	if positions are not integers, they can be set by <code>x1</code> and <code>x2</code> .
x2	if positions are not integers, they can be set by <code>x1</code> and <code>x2</code> .
gp	graphical parameters for text. It should be specified by gpar .
...	pass to grid.text . E.g. you can set text justification by just here.

Value

A data frame which contains coordinates for text.

Author(s)

Zuguang Gu <z.gu@dkfz.de>

Examples

```
hc = HilbertCurve(1, 100, level = 4, reference = TRUE)  
x = sort(sample(100, 20))  
s = x[1:10*2 - 1]  
e = x[1:10*2]  
ir = IRanges(s, e)  
labels = sample(letters, length(ir), replace = TRUE)  
hc_text(hc, ir, labels = labels)
```

HilbertCurve

*Initialize a Hilbert curve***Description**

Initialize a Hilbert curve

Usage

```
HilbertCurve(s, e, level = 4, mode = c("normal", "pixel"),
  reference = FALSE, arrow = TRUE, zoom = NULL, newpage = TRUE,
  background = "white", title = NULL, title_gp = gpar(fontsize = 16),
  legend = list())
```

Arguments

s	position that will be mapped to the start of the Hilbert curve. It should be a positive number.
e	position that will be mapped to the end of the Hilbert curve. It should be a positive number.
level	level of the Hilbert curve. There will be 4^{level} segments in the Hilbert curve.
mode	make it like a normal R plot or write the plot directly into png file. See 'details' for explanation.
reference	whether add reference line on the plot. Only works under 'normal' mode.
arrow	whether add arrows on the reference line. Only works under 'normal' mode.
zoom	internally, position are stored as integer values. To increase the resolution of the data that maps to the Hilbert curve, the original position would be zoomed according to the range of the position and the level of Hilbert curve. E.g. if the curve visualizes data ranging from 1 to 2 but level of the curve is set to 4, the positions will be zoomed by $\sim \times 2000$ so that values like 1.5, 1.555 can be mapped to the curve with more accuracy. Proper zooming factor is calculated automatically.
newpage	whether call grid.newpage to draw on a new graphic device.
background	background color, only used under 'pixel' mode.
title	title of the plot.
title_gp	graphical parameters for title. It should be specified by gpar .
legend	a grob object or a list of grob objects.

Details

This function initializes a Hilbert curve with level `level` which corresponds to the range between `s` and `e`.

Under 'normal' mode, there is a visible Hilbert curve which plays like a folded axis and different low-level graphics can be added on according to the coordinate. It only works nice if the level of the Hilbert curve is small (say less than 6).

When the level is high (e.g. > 10), the whole 2D space will be almost completely filled by the curve and it is impossible to add or visualize e.g. points on the curve. In this case, the 'pixel' mode visualizes each tiny 'segment' as a pixel and maps values to colors. So the Hilbert curve with level 11 will generate a PNG figure with 2048x2048 resolution. This is extremely useful for visualize genomic data. E.g. If we make a Hilbert curve for human chromosome 1 with level 11, then each pixel can represent 60bp (249250621/2048/2048) which is of very high resolution.

Under 'pixel' mode, if the current device is an interactive device, every time a new layer is added, the image will be add to the interactive device as a rastered image.

Value

A `HilbertCurve-class` object.

Author(s)

Zuguang Gu <z.gu@dkfz.de>

Examples

```
HilbertCurve(1, 100, reference = TRUE)
HilbertCurve(1, 100, level = 5)
HilbertCurve(1, 100, title = "title")
require(ComplexHeatmap)
cm = ColorMapping(colors = c("red", "blue"),
  levels = c("a", "b"))
legend = color_mapping_legend(cm, plot = FALSE, title = "foo")
HilbertCurve(1, 100, title = "title", legend = legend)
```

HilbertCurve-class *The HilbertCurve class*

Description

The HilbertCurve class

Details

Hilbert curve (https://en.wikipedia.org/wiki/Hilbert_curve) is a type of space-filling curves that fold one dimensional axis into a two dimensional space, but with still keeping the locality. It has advantages to visualize data with long axis with high resolution and still keep the locality of data points.

This package aims to provide an easy and flexible way to visualize data through Hilbert curve. The implementation and example figures are based on following sources:

- <http://mkweb.bcgsc.ca/hilbert/>
- <http://corte.si/posts/code/hilbert/portrait/index.html>
- <http://bioconductor.org/packages/devel/bioc/html/HilbertVis.html>

Methods

The `HilbertCurve`-class provides following methods:

- `HilbertCurve`: constructor method;
- `hc_points,HilbertCurve-method`: add points;
- `hc_segments,HilbertCurve-method`: add lines;
- `hc_rect,HilbertCurve-method`: add rectangles;
- `hc_text,HilbertCurve-method`: add text;
- `hc_layer,HilbertCurve-method`: add layers;
- `hc_png,HilbertCurve-method`: save plot as png format.

Author(s)

Zuguang Gu <z.gu@dkfz.de>

Examples

NULL

show-HilbertCurve-method

Print the HilbertCurve object

Description

Print the HilbertCurve object

Usage

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

Arguments

object A `HilbertCurve-class` object.

Value

No value is returned.

Author(s)

Zuguang Gu <z.gu@dkfz.de>

Examples

NULL

unzoom-HilbertCurve-method

Transform zoomed positions to their original values

Description

Transform zoomed positions to their original values

Usage

```
## S4 method for signature 'HilbertCurve'  
unzoom(object, x)
```

Arguments

object	A HilbertCurve-class object.
x	positions.

Details

This is a reverse function of [zoom, HilbertCurve-method](#).

The function is used internally.

Value

A numeric vector of original positions

Author(s)

Zuguang Gu <z.gu@dkfz.de>

Examples

```
hc = HilbertCurve(1, 2)  
z = zoom(hc, 1.5)  
unzoom(hc, z)
```

zoom-HilbertCurve-method

Zoom original positions

Description

Zoom original positions

Usage

```
## S4 method for signature 'HilbertCurve'  
zoom(object, x)
```

Arguments

object	A HilbertCurve-class object.
x	positions.

Details

Internally, position are stored as integer values. To increase the resolution of the data that maps to the Hilbert curve, the original position would be zoom according to the range of the position and the level of Hilbert curve. E.g. if the curve visualizes data ranging from 1 to 2 but level of the curve is set to 4, the positions will be zoomed by $\sim x2000$ so that values link 1.5, 1.555 can be mapped to the curve with more accuracy.

The function is used internally.

Value

A numeric vector which is zoomed positions.

Author(s)

Zuguang Gu <z.gu@dkfz.de>

Examples

```
hc = HilbertCurve(1, 2)  
zoom(hc, 1.5)
```


Index

- [gpar](#), [4](#), [6](#), [8–12](#)
- [grid.newpage](#), [12](#)
- [grid.text](#), [11](#)
- [grob](#), [12](#)

- [hc_layer](#)
 - [\(hc_layer-HilbertCurve-method\)](#), [2](#)
- [hc_layer](#), HilbertCurve-method
 - [\(hc_layer-HilbertCurve-method\)](#), [2](#)
- [hc_layer-HilbertCurve-method](#), [2](#)
- [hc_level](#)
 - [\(hc_level-HilbertCurve-method\)](#), [3](#)
- [hc_level](#), HilbertCurve-method
 - [\(hc_level-HilbertCurve-method\)](#), [3](#)
- [hc_level-HilbertCurve-method](#), [3](#)
- [hc_normal_points](#)
 - [\(hc_normal_points-HilbertCurve-method\)](#), [4](#)
- [hc_normal_points](#), HilbertCurve-method
 - [\(hc_normal_points-HilbertCurve-method\)](#), [4](#)
- [hc_normal_points-HilbertCurve-method](#), [4](#)
- [hc_png](#) ([hc_png-HilbertCurve-method](#)), [5](#)
- [hc_png](#), HilbertCurve-method
 - [\(hc_png-HilbertCurve-method\)](#), [5](#)
- [hc_png-HilbertCurve-method](#), [5](#)
- [hc_points](#), [2](#), [8](#), [9](#)
- [hc_points](#)
 - [\(hc_points-HilbertCurve-method\)](#), [6](#)
- [hc_points](#), HilbertCurve-method
 - [\(hc_points-HilbertCurve-method\)](#), [6](#)
- [hc_points-HilbertCurve-method](#), [6](#)
- [hc_rect](#) ([hc_rect-HilbertCurve-method](#)), [7](#)

- [hc_rect](#), HilbertCurve-method
 - [\(hc_rect-HilbertCurve-method\)](#), [7](#)
- [hc_rect-HilbertCurve-method](#), [7](#)
- [hc_segmented_points](#)
 - [\(hc_segmented_points-HilbertCurve-method\)](#), [9](#)
- [hc_segmented_points](#), HilbertCurve-method
 - [\(hc_segmented_points-HilbertCurve-method\)](#), [9](#)
- [hc_segmented_points-HilbertCurve-method](#), [9](#)
- [hc_segments](#)
 - [\(hc_segments-HilbertCurve-method\)](#), [10](#)
- [hc_segments](#), HilbertCurve-method
 - [\(hc_segments-HilbertCurve-method\)](#), [10](#)
- [hc_segments-HilbertCurve-method](#), [10](#)
- [hc_text](#) ([hc_text-HilbertCurve-method](#)), [11](#)
- [hc_text](#), HilbertCurve-method
 - [\(hc_text-HilbertCurve-method\)](#), [11](#)
- [hc_text-HilbertCurve-method](#), [11](#)
- [HilbertCurve](#), [12](#), [14](#)
- [HilbertCurve-class](#), [13](#)

- [IRanges](#), [2](#), [4](#), [6](#), [8–11](#)

- [show](#) ([show-HilbertCurve-method](#)), [14](#)
- [show](#), HilbertCurve-method
 - [\(show-HilbertCurve-method\)](#), [14](#)
- [show-HilbertCurve-method](#), [14](#)

- [unit](#), [4](#), [6](#)
- [unzoom](#) ([unzoom-HilbertCurve-method](#)), [15](#)
- [unzoom](#), HilbertCurve-method
 - [\(unzoom-HilbertCurve-method\)](#), [15](#)

unzoom-HilbertCurve-method, [15](#)

zoom (zoom-HilbertCurve-method), [16](#)

zoom, HilbertCurve-method

(zoom-HilbertCurve-method), [16](#)

zoom-HilbertCurve-method, [16](#)