

# quantsmooth

April 20, 2011

---

Chromosome14

*Example data from several quantitative genomic methods*

---

## Description

A collection of arrays that contains data of chromosome 14 of 3 colorectal tumors. The first tumor shows 1 region of loss, the second tumor shows no aberration, while the third tumor shows loss of 1 copy of the chromosome.

**affy.cn** Copy number values of 358 probes from Affymetrix 10K genechip. Data was obtained from DChip

**affy.pos** corresponding probe positions

**bac.cn** Copy number values of 112 probes from a 1 mb spaced BAC array-CGH

**bac.pos** corresponding probe positions

**ill.cn** Copy number values of 207 probes from Illumina GoldenGate Linkage IV data

**ill.pos** corresponding probe positions

## Usage

```
data(chr14)
```

## Format

Matrices of copy number values and vectors of chromosomal probe positions

## Author(s)

Jan Oosting

---

drawSimpleChrom     *Draw chromosome-like icons*

---

### Description

This function paints chromosomal icons on an existing plot

### Usage

```
drawSimpleChrom(x, y, len = 3, width = 1, fill, col, orientation = c("h", "v"),
```

### Arguments

x	start x-position
y	start y-position
len	total length of the chromosome
width	width of the chromosome
fill	character, {"a","p","q","q[1-3]","p[1-3]"}. Events to a chromosome can be depicted by coloring "a"ll of the chromosome, the complete p or q-arm, or a sub-segment of the arms
col	color(s) of fill
orientation	either "h"orizontal or "v"ertical
centromere.size	The size of the centromere as fraction of the width

### Value

This function is executed for its side effects

### Author(s)

Jan Oosting

### Examples

```
plot(c(0,4),c(0,3),type="n",xaxt="n",yaxt="n",xlab="",ylab="")
drawSimpleChrom(2,3,fill=c("p","q3"),col=c("red","blue"),orientation="v")
```

---

`getChangedRegions` *getChangedRegions*

---

**Description**

retrieve regions of interest in a vector of intensities using quantile smoothing

**Usage**

```
getChangedRegions(intensities, positions, normalized.to=1, interval, threshold)
```

**Arguments**

<code>intensities</code>	numeric vector
<code>positions</code>	numeric vector of the same length as <code>intensities</code> . If this argument is not given the results contain the indexes of the <code>intensities</code> vector, else the values in <code>positions</code> are used. Both vectors are sorted in the order of <code>positions</code> .
<code>normalized.to</code>	numeric, reference value. Changes are compared to this value
<code>interval</code>	numeric [0,1], bandwidth around reference. If the smoothed line at the higher quantile drops below the <code>normalized.to</code> value, a deleted region is recognized, and vice versa.
<code>threshold</code>	numeric, if the median smoothed value drops below <code>normalized.to - threshold</code> , or above <code>normalized.to + threshold</code> a changed region is called
<code>minlength</code>	integer, not used currently
<code>...</code>	extra arguments for <code>quantsmooth</code> function

**Details**

This function uses `quantsmooth` to detect regions in the genome that are abnormal. If `interval` is set then a smoothed line is calculated for  $\tau = 0.5 - \text{interval}/2$ , and a region is determined as upregulated if this line is above the reference. Down regulation is determined when the smoothed line for  $\tau = 0.5 + \text{interval}/2$  is below the reference value. If `threshold` is set then a smoothed line is calculated for  $\tau = 0.5$  and up- or down regulation are determined when this line is outside the range `[normalized.t - threshold:normalized.to + threshold]`

**Value**

A data.frame with 3 columns is returned. Each row contains a region with columns `up`, `start` and `end`. `start` and `end` indicate positions in the vector of the first and last position that were up- or downregulated

**Author(s)**

Jan Oosting

**Examples**

```
data(chr14)
getChangedRegions(ill.cn[,1], ill.pos, normalized.to=2, interval=0.5)
```

---

getLambdaMin	<i>getLambdaMin</i>
--------------	---------------------

---

### Description

Test a set of smoothing parameters to find best fit to data

### Usage

```
getLambdaMin(intensities, lambdas, ...)
```

### Arguments

<code>intensities</code>	numeric vector
<code>lambdas</code>	numeric vector; see <a href="#">quantsmooth</a>
<code>...</code>	extra parameters for <code>quantsmooth.cv</code> ; currently only <code>ridge.kappa</code>

### Details

Cross validation is performed using a set of lambda values in order to find the lambda value that shows the best fit to the data.

### Value

This function returns the lambda value that has the lowest cross validation value on this dataset

### Author(s)

Jan Oosting

### See Also

[quantsmooth.cv](#)

### Examples

```
data(chr14)
lambdas<-2^seq(from=-2,to=5,by=0.25)
getLambdaMin(bac.cn[,1], lambdas)
```

---

grid.chromosome      *Draw a chromosome using the grid package*

---

### Description

A chromosome is drawn including the cytobands

### Usage

```
grid.chromosome(chrom, side = 1, units = c("cM", "bases", "ISCN"), chrom.width =
                bands = "major", legend = c("chrom", "band", "none"), cex.leg = 0.7
```

### Arguments

chrom	numeric or character, id of chromosome to plot
side	numeric [1:4], side of rectangle to draw, 4 sides, side 2 and 4 are vertical
units	character, type of unit to use
chrom.width	numeric [0,1], The width relative to the width (sides 2 and 4) or height(sides 1 and 3) of the viewport
length.out	numeric, size of native units of viewport
bands	character, draw either major or minor bands
legend	character, type of legend
cex.leg	numeric, relative size of legend text
bleach	numeric [0,1], proportion by which to bleach the chromosome
...	arguments for viewport(), especially x,y, width, and height

### Details

The chromosome is drawn within a rectangle defined by x, y, width, and height, which is pushed as a viewport. The legend is drawn within the same rectangle in the space left over by chrom.width.

### Value

This function is executed for its side effects

### Author(s)

David L Duffy ,Jan Oosting

### References

lodplot package

### See Also

[paintCytobands](#)

**Examples**

```
grid.newpage()
grid.chromosome(1, units="bases", height=0.15)
```

---

lengthChromosome     *Retrieve chromosomal length*

---

**Description**

Retrieve human chromosomal length from NCBI data

**Usage**

```
lengthChromosome(chrom, units = c("cM", "bases", "ISCN"))
```

**Arguments**

chrom	vector of chromosomal id, 1:22,X,Y
units	character, one of "cM", "bases", "ISCN"

**Value**

A vector numeric in the requested units

**Author(s)**

Jan Oosting

**Examples**

```
# Show length of chromosome 1 in several types of units
lengthChromosome(1, "cM")
lengthChromosome(1, "bases")
lengthChromosome(1, "ISCN")
```

---

numericCHR     *Conversion of chromosome IDs between numeric and character*

---

**Description**

The function converts chromosomal ids to their numeric form, and the sex chromosomes to values between 98 and 100. This simplifies sorting on chromosome ID

**Usage**

```
numericCHR(CHR)
characterCHR(CHR)
```

**Arguments**

CHR character/numeric vector for both functions the mode of the input is not forced. For numericCHR strings "X","Y" and "XY" are converted to 98,99 and 100 respectively.

**Value**

numericCHR returns a numeric vector of same length as CHR characterCHR returns a character vector of same length as CHR

**Author(s)**

Jan Oosting

**Examples**

```
chroms<-c("3","2","8","X","7","Y","5","1","9","10","11","12","4","6")
sort(chroms)
sort(numericCHR(chroms))
characterCHR(sort(numericCHR(chroms)))
```

---

paintCytobands

*Paint a chromosomal idiogram*

---

**Description**

Paints a human chromosomal idiogram in an existing plot Adapted from the paint.chromosome function in the lodplot package by David L Duffy

**Usage**

```
paintCytobands(chrom, pos = c(0, 0), units = c("cM", "bases", "ISCN"), width = 0,
length.out, bands = "major", orientation = c("h", "v"), legend = TRUE,
cex.leg = 0.7, bleach = 0, ...)
```

**Arguments**

chrom chromosomal id, chromosome to plot 1:22,X,Y

pos numeric vector of length 2, position in the plot to start the plot

units units along which to plot the chromosome

width numeric, width of the chromosome, the chromosome is plotted between pos[2] and pos[2]-width

length.out numeric, if given, the chromosome will have this length in the plot

bands if not equal to "major", then also the minor bands will be plotted

orientation chromosome is plotted either *Horizontally* to the right of the starting point or *Vertically* down from the starting point

legend logical, if TRUE then the bandnames are plotted next to the chromosome

cex.leg numeric, relative size of legend text

bleach numeric [0,1], proportion by which to bleach the chromosome

... extra parameters for plot

**Value**

This function is executed for its side effects

**Author(s)**

David L Duffy , Jan Oosting

**References**

lodplot package

**Examples**

```
plot(c(0, lengthChromosome(14, "bases")), c(-2, 2), type="n", xaxt="n", yaxt="n", xlab="", ylab="")
paintCytobands(14, units="bases")
```

---

plotChromosome      *Wrapper for plotSmoothed*

---

**Description**

This function is a wrapper for plotSmoothed, to make data subsetting easier

**Usage**

```
plotChromosome(gendata, chrompos, chromosome, dataselection = NULL, ylim = NULL,
```

**Arguments**

gendata	numeric matrix or data.frame
chrompos	chrompos object with same numer of rows as gendata
chromosome	numeric, chromosme to show
dataselection	optional, subset of samples/columns in gendata
ylim	limits for plot
normalized.to	y-value(s) for line
grid	x-value(s) for line
smooth.lambda	smoothing parameter, see <a href="#">quantsmooth</a>
interval	position of extra lines besides median, see <a href="#">plotSmoothed</a>
...	extra arguments for <a href="#">plotSmoothed</a>

**Value**

The function is used for its side effects



**Author(s)**

Jan Oosting

**See Also**[plotSmoothed](#), [quantsmooth](#)


---

plotSmoothed	<i>plotSmoothed</i>
--------------	---------------------

---

**Description**

Plot a smoothed line together with the original data values

**Usage**

```
plotSmoothed(intensities, position, ylim=NULL, ylab="intensity", xlab="position")
```

**Arguments**

<code>intensities</code>	numeric vector or matrix, data are plotted by column
<code>position</code>	numeric vector; the length should be the number of rows in <code>intensities</code>
<code>ylim</code>	numeric vector of length 2, limits for plot. If <code>NULL</code> then the minimal and maximal value in <code>intensities</code> is used
<code>ylab</code>	character, label for y-position
<code>xlab</code>	character, label for x-position
<code>normalized.to</code>	numeric, a line(s) is drawn at this horizontal position
<code>grid</code>	numeric, a line(s) is drawn at this vertical position
<code>smooth.lambda</code>	numeric, smoothing parameter see <code>quantsmooth</code>
<code>interval</code>	numeric (0..1), plotting of extra smoothed lines around median. With <code>interval = 0.5</code> the 0.25 and 0.75 quartiles are plotted, with <code>interval = 0.9</code> the 0.05 and 0.95 quantiles are plotted,
<code>plotnew</code>	logical, if <code>TRUE</code> a new plot is created, else the data are plotted into an existing plot
<code>cols</code>	color vector, colors for columns in <code>intensities</code>
<code>cex.pts</code>	size of the dots in the plot. Set to 0 to skip plotting the dots
<code>...</code>	extra parameters for <code>plot</code>

**Details**

This function plots the raw data values as dots and the median smoothed values as a continuous line. If `interval` is supplied these are plotted as lines in different line types. More than 1 `interval` can be given.

**Value**

This function is used for its side effects

**Author(s)**

Jan Oosting

**See Also**

[quantsmooth](#)

**Examples**

```
data(chr14)
plotSmoothed(bac.cn,bac.pos,ylim=c(1,2.5),normalized.to=2,smooth.lambda=2.5)
```

---

`position2Cytoband` *Determine cytoband position based on location of probe*

---

**Description**

Determine cytoband position based on location of probe

**Usage**

```
position2Cytoband(chrom, position, units = c("cM", "bases", "ISCN"), bands = c("
```

**Arguments**

<code>chrom</code>	chromosomal id, chromosome to plot 1:22,X,Y
<code>position</code>	numeric vector
<code>units</code>	character, type of positional unit
<code>bands</code>	character, type of cytoband

**Value**

Character vector with cytobands, if an illegal position was used, the value "-" is returned. All positions within a single function call should be for a single chromosome

**Author(s)**

Jan Oosting

**See Also**

[lengthChromosome](#)

**Examples**

```
position2Cytoband(1,c(50e6,125e6,200e6),units="bases")
position2Cytoband(1,c(50,125,200),units="cM",bands="minor")
```

---

prepareGenomePlot *Set up a full genome plot*

---

### Description

This function starts up a plot consisting of all chromosomes of a genomen, including axes with chromosome names.

### Usage

```
prepareGenomePlot(chrompos, cols = "grey50", paintCytobands = FALSE, bleach = 0,  
                  sexChromosomes = FALSE, units = c("bases", "cM", "ISCN"),...)
```

### Arguments

chrompos	chrompos object, data.frame with CHR column identifying the chromosome of probes, and a MapInfo column identifying the position on the chromosome
cols	color(s) for the chromosome lines
paintCytobands	logical, use paintCytoband to plot ideograms for all chromosomes
bleach	numeric [0,1], proportion by which to bleach the ideograms
topspace	numerical, extra space on top of plot, i.e. for legends
organism	character, if given a 2 column plot is created with the chromosomes for the given species. Currently "hsa", "mmu", and "rno" are supported
sexChromosomes	logical, if TRUE then also the sex chromosomes X and Y are plotted
units	character, type of units for genomic data
...	extra arguments for plot function

### Details

If organism is not supplied then a single column is plotted of the available chromosomes in chrompos\$CHR. The arguments paintCytobands, bleach, and sexChromosomes are not used in that case.

### Value

A matrix with 2 columns that contain the Y and X positions for the probes on the plot

### Author(s)

Jan Oosting

---

quantsmooth.cv      *quantsmooth.cv*

---

## Description

Cross validation of smoothing parameters

## Usage

```
quantsmooth.cv(intensities, smooth.lambda=2, ridge.kappa=0)
```

## Arguments

`intensities`    numeric vector  
`smooth.lambda`  
                  numeric; see [quantsmooth](#)  
`ridge.kappa`    fudge parameter; see [quantsmooth](#)

## Details

Cross validation is performed by calculating the fit from the even indices on the odd indices and vice versa.

## Value

This function returns the sum of squared differences or NA if the fitting function gave an error

## Author(s)

Jan Oosting

## See Also

[getLambdaMin](#)

## Examples

```
data(chr14)
# A low value is indicative of a better fit to the data
quantsmooth.cv(bac.cn[,1],1)
quantsmooth.cv(bac.cn[,1],2.8)
```

---

quantsmooth      *quantsmooth*

---

**Description**

Quantile smoothing of array data

**Usage**

```
quantsmooth(intensities, smooth.lambda=2, tau=0.5, ridge.kappa=0, smooth.na=TRUE)
```

**Arguments**

`intensities`    numeric vector  
`smooth.lambda`    numeric  
`tau`            numeric [0..1], the quantile desired; see [rq.fit](#)  
`ridge.kappa`    fudge parameter; see details  
`smooth.na`      logical; handling of NA  
`segment`       integer, length of overlapping segments

**Value**

This function returns a vector of the same length as `intensities`, or a matrix if the length of `tau` is greater than 1.

**Author(s)**

Jan Oosting

**Examples**

```
data(chr14)
plot(quantsmooth(bac.cn[,1], smooth.lambda=2.8), type="l")
```

---

quantsmooth.seg      *quantsmooth.seg*

---

**Description**

segmented Quantile smoothing of array data

**Usage**

```
quantsmooth.seg(y, x = 1:length(y), lambda = 2, tau = 0.5, kappa = 0, nb = leng
```

**Arguments**

<code>y</code>	numeric vector
<code>x</code>	numeric vector of same length as <code>y</code> . Position of values
<code>lambda</code>	numeric
<code>tau</code>	numeric [0..1], the quantile desired; see <a href="#">rq.fit</a>
<code>kappa</code>	fudge parameter; see details
<code>nb</code>	integer, basis

**Value**

This function returns a vector of the same length as `y`

**Author(s)**

Jan Oosting

**Examples**

```
data(chr14)
plot(quantsmooth.seg(bac.cn[,1], lambda=2.8, nb=50), type="l")
```

---

scaleto

*Scales data within a range to a new range*

---

**Description**

This function scales data to a new range while enforcing the boundaries. This can be helpful in preventing overlap between chromosomal plots that display multiple chromosomes in the same plot

**Usage**

```
scaleto(x, fromlimits = c(0, 50), tolimits = c(0.5, -0.5), adjust = TRUE)
```

**Arguments**

<code>x</code>	numeric
<code>fromlimits</code>	numeric vector with length 2, original range of data
<code>tolimits</code>	numeric vector with length 2, target range of data
<code>adjust</code>	logical, if TRUE then the target values are clipped to the target range

**Value**

numeric of same size as `x`

**Author(s)**

Jan Oosting

# Index

- \*Topic **aplot**
  - drawSimpleChrom, 2
  - grid.chromosome, 5
  - paintCytobands, 7
- \*Topic **attribute**
  - getChangedRegions, 3
- \*Topic **datasets**
  - Chromosome14, 1
- \*Topic **data**
  - lengthChromosome, 6
- \*Topic **hplot**
  - plotChromosome, 8
  - plotSmoothed, 9
  - prepareGenomePlot, 11
- \*Topic **htest**
  - quantsmooth.cv, 12
- \*Topic **manip**
  - numericCHR, 6
  - position2Cytoband, 10
  - scaleto, 14
- \*Topic **smooth**
  - getLambdaMin, 4
  - quantsmooth, 13
  - quantsmooth.cv, 12
  - quantsmooth.seg, 13
  
- affy.cn (*Chromosome14*), 1
- affy.pos (*Chromosome14*), 1
  
- bac.cn (*Chromosome14*), 1
- bac.pos (*Chromosome14*), 1
  
- characterCHR (*numericCHR*), 6
- chr14 (*Chromosome14*), 1
- Chromosome14, 1
  
- drawSimpleChrom, 2
  
- getChangedRegions, 3
- getLambdaMin, 4, 12
- grid.chromosome, 5
  
- ill.cn (*Chromosome14*), 1
- ill.pos (*Chromosome14*), 1
  
- lengthChromosome, 6, 10
  
- numericCHR, 6
  
- paintCytobands, 5, 7
- plotChromosome, 8
- plotSmoothed, 8, 9, 9
- position2Cytoband, 10
- prepareGenomePlot, 11
  
- quantsmooth, 4, 8–10, 12, 13
- quantsmooth.cv, 4, 12
- quantsmooth.seg, 13
  
- rq.fit, 13, 14
  
- scaleto, 14