

Genome Visualization with Circos

Session 1 — Displaying and Formatting Ideograms

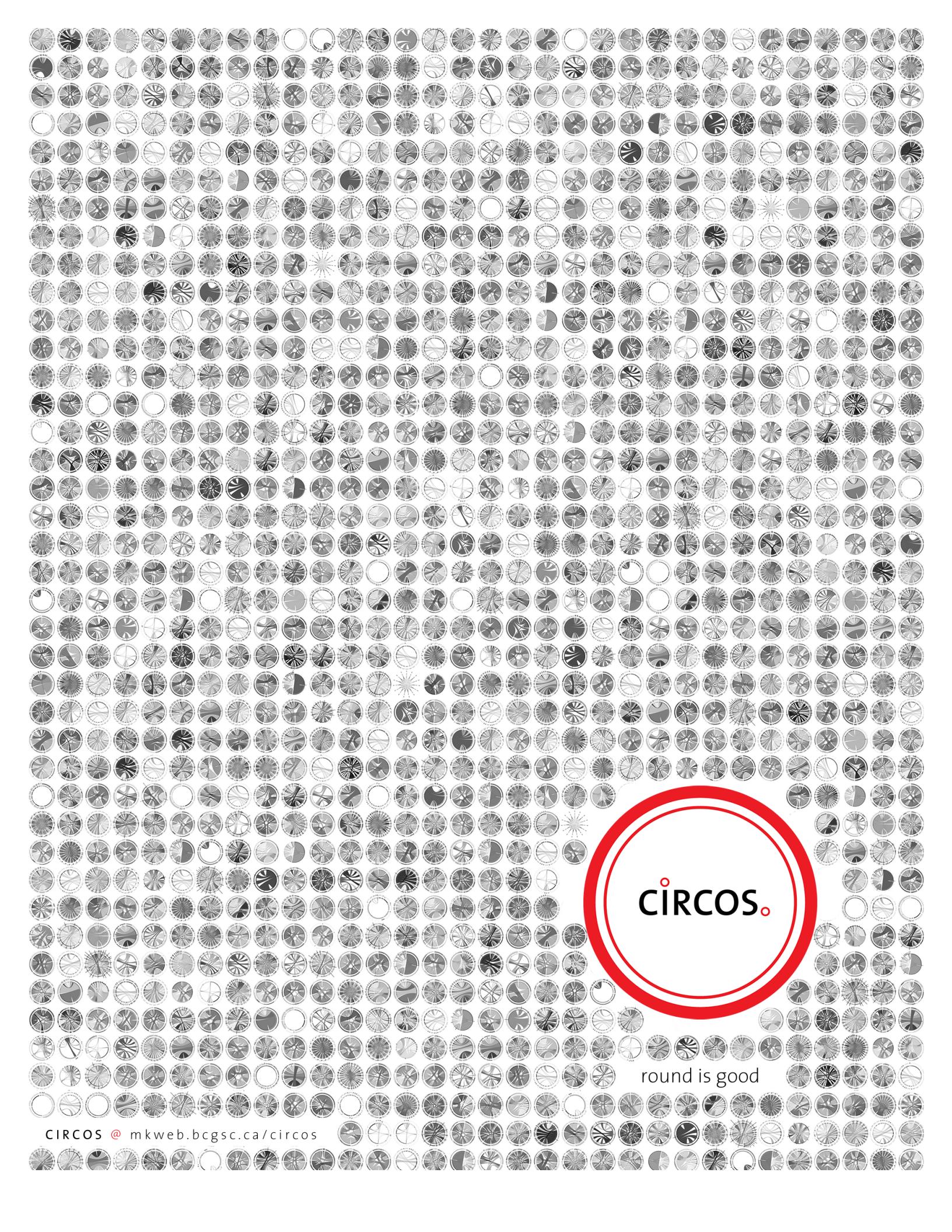
Martin Krzywinski
Genome Sciences Centre
100-570 West 7th Ave
Vancouver BC V5Z 4S6 Canada
1-604-877-6000 x 673262
martink@bcgsc.ca
<http://mkweb.bcgsc.ca>

Circos
<http://circos.ca>

Course Materials
<http://circos.ca/documentation/course>

Genome Sciences Center
<http://bcgsc.ca>

Version History
v0.22 3 May 2016
v0.21 12 May 2014
v0.20 6 May 2014
v0.19 21 Apr 2014
v0.18 7 May 2012
v0.17 5 Jun 2011
v0.16 23 Jul 2010
v0.15 12 Jul 2010
v0.14 30 Jun 2010
v0.13 30 Jun 2010
v0.12 29 Jun 2010
v0.11 29 Jun 2010
v0.10 15 Jun 2010



CIRCOS.

round is good

CIRCOS @ mkweb.bcgsc.ca/circos

Table of Contents

Table of Contents.....	1
Introduction to Ideogram Layout	3
What You Will Learn	3
Lesson 1 – Drawing and Spacing Ideograms	4
Defining a Karyotype.....	4
Generating the Image	5
Spacing Between Ideograms.....	6
Spacing Between Specific Pairs of Ideograms.....	7
Lesson 2 – Relative Ideogram Spacing	8
Benefits of Relative Spacing.....	9
Lesson 3 – Changing Ideogram Scale.....	10
Relative Scale.....	10
Relative Scale with Normalization	11
Lesson 4 – Ideogram Selection.....	12
Lesson 5 – Ideogram Order.....	13
Lesson 6 – Drawing Ideogram Regions	14
Ideogram vs Chromosome.....	14
Axis Breaks	14
Lesson 7 – Chromosome Breaks.....	15
Lesson 8 – Ordering Ideogram Regions.....	17
Lesson 9 – Cytogenetic Bands.....	18
Transparent Colors.....	18
Lesson 10 – Drawing Multiple Genomes	19
Chromosome Color Scheme	20
Luminance Normalized Colors	21

Lesson 11 – Ideogram, Color, Scale, Order and Orientation.....	23
Filtering Ideograms	23
Ideogram Color.....	24
Ideogram Progression.....	24
Relative Ideogram Scaling	24
Symmetric Ideogram Layout.....	25
Lesson 12 – Relative and Absolute Ticks	26
Figures.....	30

Introduction to Ideogram Layout

Central in a Circos figure are the *ideograms* and their order, scale and orientation. An ideogram is the graphical depiction of a chromosome. A chromosome may be shown as a single ideogram, in whole or cropped, or as multiple ideograms, if you divide it into several regions. It is important to create an ideogram layout that helps the reader parse and understand the data.

For example, if you are comparing a single chromosome of one genome (e.g. human chr1) to an entire mammalian genome (e.g. mouse), it is helpful to magnify the human chromosome (e.g. so that it takes 50% of the figure) to present its data at a higher resolution. On the other hand, if you are comparing two genomes, or roughly equal size subsets of two genomes, it is helpful to scale the ideograms to have the same size to create a pleasing symmetrical layout.

WHAT YOU WILL LEARN

In this first practical session you will learn how to position, order, crop and format the ideograms.

In subsequent sessions you will learn about data tracks, how to place and format them, and how to write rules that dynamically change how data points are shown.

Lesson 1 – Drawing and Spacing Ideograms

Let's take a look at the configuration file for this lesson. Subsequent lessons will build on this file, therefore it is important to understand all of its components.

```
# the karyotype parameter specifies the file which defines the
# size and name of each chromosome for the figure
karyotype = ../../data/karyotype.5chr.txt

# unit of length for the chromosomes - this is used
# in other parts of the file where a position is referenced
chromosomes_units = 1000000

# toggle to display all of the chromosomes in the
# karyotype file, in the order of appearance
chromosomes_display_default = yes

# spacing, position, thickness, color/outline format of the ideograms
# from same directory as circos.conf
<<include ideogram.conf>>

# position, labels, spacing and grids for tick marks
# from shared session directory
<<include ../../etc/ticks.conf>>

# size, format and location of the output image, shared by
# all the sessions in this course
<<include ../../etc/image.conf>>

# the files below are included from the Circos distribution
# DO NOT REMOVE THESE
# defines colors, fonts and fill patterns
<<include etc/colors_fonts_patterns.conf>>

# the housekeeping file contains system parameters that
# define file delimiters, debugging options and other global settings
<<include etc/housekeeping.conf>>
```

DEFINING A KARYOTYPE

For this lesson, we'll focus on the karyotype file, which defines the name, size and color of chromosomes. Any subset of these chromosomes (or regions thereof) can be drawn. In this example, we'll define 5 chromosomes named `chr1` ... `chr5` of sizes 5, 10, 20, 50 and 100 Mb. Once this file is defined for a specific genome, you don't need to edit it again (unless the chromosome size and number changes with a new assembly). Karyotypes for common organisms (human, mouse, rat, chimp, fly, etc.) are in `data/karyotype` in the Circos distribution.

The colors used in this file must be defined in the `<colors>` block in the configuration file (this block is defined in the `etc/colors_fonts_patterns.conf` file, which is imported). Colors are predefined for each hue (e.g. `vdred`, `dred`, `red`, `lred`, `vlred`, `v=very`, `d=dark`, `l=light`) as well as Brewer palettes (<http://www.colorbrewer.org>). The named red colors actually point to Brewer colors

```
vvlred = reds-7-seq-1 # very very light red
vlred = reds-7-seq-2 #      very light red
lred = reds-7-seq-3 #          light red
red = reds-7-seq-4 #              red
dred = reds-7-seq-5 #      dark red
vdred = reds-7-seq-6 #      very dark red
vvdred = reds-7-seq-7 # very very dark red
```

For chromosomes in this lesson we will use the 5 color spectral Brewer palette (`spectral-5-div-{1-5}`) (Figure 1).

The karyotype file for this lesson is found in `1/data/karyotype.5chr.txt`

```
# A simple karyotype with 5 chromosomes:
#
# chr1 5Mb
# chr2 10Mb
# chr3 20Mb
# chr4 50Mb
# chr5 100Mb
#
# The format of this file is
#
# chr - CHRNAME CHRLABEL START END COLOR
#
# In data files, chromosomes are referred to by CHRNAME.
# On the image, they are labeled by CHRLABEL
#
# Colors are taken from the spectral Brewer palette.
# To learn about Brewer palettes, see
#
# www.colorbrewer2.org
# mkweb.bcgsc.ca/brewer

chr - chr1 1 0 5000000 spectral-5-div-1
chr - chr2 2 0 10000000 spectral-5-div-2
chr - chr3 3 0 20000000 spectral-5-div-3
chr - chr4 4 0 50000000 spectral-5-div-4
chr - chr5 5 0 100000000 spectral-5-div-5
```

GENERATING THE IMAGE

To generate the image, run Circos using this lessons configuration file.

```
> cd 1/1
> circos
debuggroup summary 0.22s welcome to circos v0.69 6 Dec 2015 on Perl 5.010000
debuggroup summary 0.22s current working directory
/home/martink/work/circos/course/3/session/1/1
debuggroup summary 0.22s command /home/martink/bin/circos [no flags]
debuggroup summary 0.22s guessing configuration file
debuggroup summary 0.23s found conf file
/home/martink/work/circos/course/3/session/1/1/etc/circos.conf
debuggroup summary 0.51s debug will appear for these features: output,summary
```

```
debuggroup summary 0.51s bitmap output image ./circos.png
debuggroup summary 0.51s parsing karyotype and organizing ideograms
debuggroup summary 0.52s karyotype has 5 chromosomes of total size 185,000,005
debuggroup summary 0.52s applying global and local scaling
debuggroup summary 0.52s allocating image, colors and brushes
debuggroup summary 1.47s drawing 5 ideograms of total size 185,000,005
debuggroup summary 1.47s drawing highlights and ideograms
debuggroup output 1.62s generating output
debuggroup output 1.72s created PNG image ./circos.png (62 kb)
```

Circos will generate some output that describes what it is doing and the location of the output image. The image is shown in Figure 2.

SPACING BETWEEN IDEOGRAMS

Spacing between ideograms is controlled in the `<ideogram>` block, by variables in the `<spacing>` block. The `<ideogram>` block is added to the `circos.conf` configuration file through the `<<include ideogram.conf>>` directive. The relevant parts of the `ideogram.conf` file for this lesson are

```
# 2/1/etc/ideogram.conf
<ideogram>
  <spacing>
    default = 2u
    #default = 10u
    #<pairwise chr1>
    #spacing = 5u
    #</pairwise>
    #<pairwise chr2 chr3>
    #spacing = 15u
    #</pairwise>
    #<pairwise chr3 chr4>
    #spacing = 25u
    #</pairwise>
  </spacing>
  ...
</ideogram>
```

In this example, the spacing between ideograms is `default = 2u`. The `u` suffix indicates that the units of this value are given relative to the value defined by `chromosomes_units`. Since `chromosomes_units = 1000000`, the spacing is 2000000 (2 Mb). To increase spacing, uncomment the `default = 10u` line (don't forget to comment out the original `default = 2u` line).

```
<spacing>
#default = 2u
default = 10u
```

The result of this change is shown in Figure 3.

SPACING BETWEEN SPECIFIC PAIRS OF IDEOGRAMS

Spacing between individual ideogram pairs can be adjusted using `<pairwise>` blocks within the `<spacing>` block. For each pair, a different `<pairwise>` block is defined, with the name of the block given by “`name1 name2`” where `name1` and `name2` are the names of the adjacent chromosomes. You can also use `name1, name2` as alternative syntax. If you specify only one ideogram in the `<pairwise>` block, spacing on either side of it will be changed.

```
# 2/1/etc/ideogram.conf
<ideogram>
<spacing>
#default = 2u
default = 10u
<pairwise chr1>
spacing = 5u
</pairwise>
#<pairwise chr2 chr3>
#spacing = 15u
#</pairwise>
#<pairwise chr3 chr4>
#spacing = 25u
#</pairwise>
</spacing>
...
</ideogram>
```

By uncommenting the first `<pairwise>` block, the space adjacent to chromosomes 1 is changed from `10u` (which is the new default) to `5u`. The adjusted spacing is shown in Figure 4.

By commenting out the remaining `<pairwise>` blocks, the spacing between chromosomes 2 and 3 is increased to `15u` and between 3 and 4 to `25u`. The result is shown in Figure 5.

```
# 2/1/etc/ideogram.conf
<ideogram>
<spacing>
#default = 2u
default = 10u
<pairwise chr1>
spacing = 5u
</pairwise>
<pairwise chr2 chr3>
spacing = 15u
</pairwise>
<pairwise chr3 chr4>
spacing = 25u
</pairwise>
</spacing>
...
</ideogram>
```

Lesson 2 – Relative Ideogram Spacing

In the previous lesson we have defined the spacing using the `u` suffix. This sets the spacing to be an absolute value, relative to the value of `chromosomes_units`. Thus, for a spacing value of `2u`, the spacing will always be 2 `chromosomes_units` (2 Mb in this example), regardless of the size of the ideograms (see Figure 2).

By changing the suffix from `u` to `r`, you can set the spacing to be relative to the total size of all displayed ideograms. This is a useful feature when you don't know what the ideogram size is, but you would like to keep the same visual spacing layout in the figure.

By uncommenting `default=0.1r`, spacing between ideograms is set to 18.5 Mb (10% of total ideogram size, which is 185 Mb). The result is shown in Figure 6.

```
# 2/1/etc/ideogram.conf

<ideogram>
<spacing>

#default = 2u

# When spacing unit is "r", the fraction is calculated
# relative to the total size of all ideograms.
# e.g. if all ideograms total 185 Mb, 0.1r spacing is 18.5Mb

default = 0.1r

...
</ideogram>
```

Absolute and relative spacing can be mixed. In the configuration file below, the default spacing between ideograms is `0.1r` (10% of total ideogram size), but spacing between chromosomes 1 & 2, 2 & 3 and 3 & 4 is set to absolute values of `0u`, `1u` and `2r`, respectively. When pairwise spacing is defined relatively (e.g. `2r` for chromosomes 3 & 4), it is relative to the default spacing. The result is shown in Figure 7.

```
<spacing>

#default = 2u

# When spacing unit is "r", the fraction is calculated
# relative to the total size of all ideograms.
# e.g. if all ideograms total 185 Mb, 0.1r spacing is 18.5Mb

default = 0.1r

<pairwise chr1 chr2>
spacing = 0u # no space
</pairwise>

<pairwise chr2 chr3>
spacing = 2u # 2Mb space
```

```
</pairwise>

<pairwise chr3 chr4>
spacing = 2r # 2x default space
</pairwise>

</spacing>
```

BENEFITS OF RELATIVE SPACING

Relative spacing is useful when you don't know the size of the ideograms, or are changing their scale (covered in another lesson). Absolute spacing, on the other hand, is useful when ideogram size is known. You can mix and match these spacing types freely.

If you are displaying a figure of an entire genome (e.g. 24 ideograms of the human genome) you may want to dedicate approximately 25% of the ideogram circle's circumference to spacing (if so, use `0.01r` to set the spacing between each ideogram to 1%). On the other hand, it may make more sense to you to put the equivalent of 20Mb of space between each ideogram (if so, use `20u`, assuming `chromosomes_units = 1000000`).

Lesson 3 – Changing Ideogram Scale

Just as you can adjust the space between ideograms, you can adjust the scale of the ideogram itself in order to adjust its size in the figure. This change of scale is useful if you wish to display an ideogram larger than physical scale (e.g. show chromosome 1 at 10× magnification), or if you wish to reduce the importance of an ideogram (e.g. show chromosome 2 at 5× reduction).

Scale adjustment is defined in `circos.conf` using the `chromosomes_scale` parameter. The syntax of the value for this parameter is

```
chromosomes_scale = chr1=0.2,chr2=2,chr3=10
```

which would display chromosome 1 at 0.2× magnification (5× reduction), chromosome 2 at 2× magnification and chromosome 3 at 10× magnification, respectively.

Take a look at `session/2/3/etc/circos.conf` and uncomment the first `chromosomes_scale` line. This will set chromosome 1 to 0.5× magnification (2× reduction).

```
chromosomes_scale = chr1=0.5
#chromosomes_scale = chr1=0.5,chr2=2,chr3=10
# chr1 occupies 50% of figure
#chromosomes_scale = chr1=0.5r

# chr5 occupies 25% of figure
# chr4 occupies 25% of figure
#chromosomes_scale = chr5=0.25r,chr4=0.25r
```

Recall that chromosome 1 was defined in the karyotype file to be 5 Mb in size and chromosome 2 to 10 Mb in size. When the scale for the ideograms was unaltered (by default all ideograms are displayed at 1×), chromosome 1 appeared to be $\frac{1}{2}$ of the size of chromosome 2 (because its length was $\frac{1}{2}$ of chromosome 2). Now that we have changed the scale of chromosome 1 to 0.5× it appears to be $\frac{1}{4}$ of the size of chromosome 2. This can be seen in Figure 8.

If you uncomment the second `chromosomes_scale` line

```
# chromosomes_scale = chr1=0.5
chromosomes_scale = chr1=0.5,chr2=2,chr3=10
```

in addition to shrinking chromosome 1, the size of chromosomes 2 and 3 will be changed to 2× and 10× original size, respectively (Figure 9).

RELATIVE SCALE

A common requirement is to have a figure in which an ideogram (or subset of ideograms) occupy some fraction of the total figure. For example, suppose you wanted chromosome 1 to occupy 50% of the figure. To do this, simply specify the scale as the fraction of the figure you wish the chromosome to occupy and use the `r` suffix to indicate that the scale is relative (to the size of the figure).

```
# chr1 occupies 50% of figure
chromosomes_scale = chr1=0.5r
```

yields the required result in Figure 10.

Given that chromosomes 4 and 5 are quite large (50 and 100 Mb, respectively), let's adjust their scale to occupy 50% of the image, with each ideogram occupying 25% of the image.

```
# chr5 occupies 25% of figure
# chr4 occupies 25% of figure
chromosomes_scale = chr5=0.25r,chr4=0.25r
```

produces Figure 11.

You've probably noticed that as a result of the change of scale, the tick marks are getting crowded and their labels are starting to overlap. Formatting tick marks will be covered in another lesson.

RELATIVE SCALE WITH NORMALIZATION

You can use regular expressions to apply scaling to several ideograms at the same time

```
chromosomes_scale = /chr[123]/=0.5rn
```

This will apply scale to chr1, chr2 and chr3, which are the chromosome names that match the regular expression. The special `rn` suffix indicates normalized relative scale. Half of the image (`0.5r`) will be assigned to the m chromosomes that match the regular expression, and each chromosome will occupy $0.5/m$ of the image. For the present example, the effect is to divide 0.5 of the figure *evenly* among the three chromosomes (Figure 12). This can be done manually, but you'd have to calculate the exact scale factor for each chromosome because they all have different size.

Finally, if you want to set all the chromosomes to appear to be the same size,

```
chromosomes_scale = ./.=1rn
```

The regular expression `.` matches everything, and the effect will be to divide the full figure (hence `1rn`) evenly among the chromosomes (Figure 13). The expression `./.=0.2r` would have achieved the same thing (since we have 5 ideograms), but would no longer work if you added more ideograms to the figure. By using `1rn`, you don't need to remember how many ideograms you are drawing.

This lesson demonstrated how to change the *global* scale of an ideogram. *Global* means that the all regions of the ideogram are magnified (or reduced) uniformly. Circos allows you to adjust the scale of a region of an ideogram and vary the scale smoothly across the ideogram. You can define multiple such local scale adjustments.

Lesson 4 – Ideogram Selection

Up to now, all of the chromosomes that were defined in the karyotype were shown. This is the default behaviour, explicitly defined by

```
# toggle to display all of the chromosomes in the
# karyotype file, in the order of appearance
chromosomes_display_default = yes
```

in the `circos.conf` file. This is what is shown in Figure 2.

To draw a subset of chromosomes, you need to change `chromosomes_display_default` to no and set the `chromosomes` parameter to list the chromosomes you wish to show. For example, to show only chromosomes 1, 2 and 3

```
#chromosomes_display_default = yes

chromosomes_display_default = no
chromosomes = chr1;chr2;chr3
```

The result is shown in Figure 14. You will notice that the spacing between ideograms in this figure looks quite different than in Figure 2. The spacing has not changed (`2u`) but because the total ideogram size has decreased (from 185 Mb when all ideograms were shown to 35 Mb when only chromosomes 1, 2 and 3 are shown), a `2u` space now looks quite big.

To cope with this (undesired) increase in spacing, use a relative ideogram spacing. Relative spacing ensures that the fraction of the ideogram circle occupied by spacing is the same, regardless of the total size of ideogram in the figure. Figure 15 shows the effect of using `0.01r` spacing on two figures with 35 Mb and 185 Mb of ideograms. The spacing is adjusted in the `<ideogram>` block that is conventionally found in `ideogram.conf`.

Lesson 5 – Ideogram Order

By default, ideograms are placed in the figure in the same order as their chromosomes appear in the karyotype file. It is therefore sensible to define your karyotype file with a chromosome order that is a reasonable default (e.g. increasing by chromosome number).

To change the ideogram order, use the `chromosomes_order` parameter. To change the order of chromosomes 3, 4 and 5, uncomment the first `chromosomes_order` line.

```
# explicitly define order
chromosomes_order = chr1,chr2,chr5,chr4,chr3

# relative order
#chromosomes_order = chr3,chr5

# relative order
#chromosomes_order = chr1,chr4,-,chr3,chr5
```

The change in order is shown in Figure 16.

If you have a large number of chromosomes but only need to adjust the order of a few ideograms, it is not convenient to have to specify the order of the entire set. Instead, you can set the order to include only the ideograms you wish to reorder.

```
# relative order
chromosomes_order = chr3,chr5
```

Only the order of chromosomes 3 and 5 is affected. In this case, when the value of the parameter lists a subset of chromosomes, the first chromosome in the list acts as an anchor (its order is not affected – it appears in the same position as it would by default) and any subsequent chromosomes listed in the value appear next. Thus, in this case chromosome 3 acts as the anchor (its order is not changed) and is followed immediately by chromosome 5. Once chromosome 5 is placed, only chromosome 4 remains to be drawn and thus chromosome 4 appears last. This is shown in Figure 17.

You can reorder several subsets of ideograms by separating the order of subsets by “-”, one for each unordered ideogram. Figure 18 shows the effect of changing the order parameter to

```
# relative order
chromosomes_order = chr1,chr4,-,-,chr2
```

In this case there are two subsets that are reordered: chromosomes 1,4 and chromosomes 2. In the first subset chromosome 1 acts as the anchor (it is placed first, as it would by default) and followed by chromosome 4. The last chromosome is chromosome 2 and the two “-“ correspond to the remaining chromosomes whose order has not been explicitly stated (chromosomes 3 and 5).

Lesson 6 – Drawing Ideogram Regions

So far, the ideogram of each chromosome showed the entire chromosome. Thus, if chromosome 1 was defined to have a start of 0 and an end of 5000000 in the karyotype file, its ideogram corresponded to the full extent of chromosome 1.

IDEOGRAM VS CHROMOSOME

This lesson emphasizes the difference between a *chromosome* and an *ideogram*. In Circos, the *chromosome* is the structure defined in the karyotype file. Normally, this structure would correspond to an entire chromosome, sequence contig, or any other ordered scale. The *ideogram*, on the other hand, is the graphical representation of the chromosome in the figure. You'll see later that a chromosome can be represented by *multiple ideograms*, through the use of axis breaks.

AXIS BREAKS

To change the region of the chromosome that is drawn, the `chromosomes` parameter is used. Recall that this is the same parameter used to select ideograms for display, as seen in a previous lesson. To draw 9-21 Mb of chromosome 5, the `chromosomes` parameter is set to `chr5:9-21`, as shown below.

```
chromosomes_units = 1000000
chromosomes_display_default = yes

chromosomes = chr5:9-21
```

Notice two things here. First, the region of chromosome 5 is expressed in units of `chromosomes_units`, to avoid having many non-significant zeroes in the configuration file. Since the unit is 1 Mb, the range is 9-21 Mb.

Second, notice that unlike in the previous lesson in which a subset of chromosomes was selected for display, in this case `chromosomes_display_default=yes`. The `yes` value of this parameter results in all chromosomes being drawn. In this case, the `chromosomes` parameter is used to adjust the region of one or more chromosomes. Chromosomes 1-4 are drawn in their entirety (`chromosomes_display_default=yes` ensures that all chromosomes that are not explicitly removed from display are drawn) and chromosome 5 is cropped.

The result of cropping chromosome 5 is shown in Figure 19.

You can crop any number of chromosomes in this way. For example, settings `chromosomes` to

```
chromosomes = chr3:8-12;chr4:4-11;chr5:9-21
```

affects the ideogram extent of chromosomes 3, 4 and 5. This is shown in Figure 20.

Lesson 7 – Chromosome Breaks

The previous lesson showed you how to crop the ideogram of a chromosome to a region. This kind of cropping is very useful if you know what part of a chromosome you wish to show.

On the other hand, if you know what part of a chromosome you *don't* want to show, use a chromosome break. A break is defined analogously to a crop, except that the parameter is `chromosomes_breaks` and the chromosome and region combination to remove is preceded by a `"-"`.

For example, to remove the 11-19 Mb region of chromosome 5 from the display, set

```
chromosomes_breaks = -chr5:11-19
```

In place of the region, an axis break is shown. The result is shown in Figure 21.

The format of the axis break is controlled in the `<ideogram><spacing>` block. The `break` parameter defines the size of the break, which, like spacing, can be set to absolute or relative. When the `break` value is relative, it is relative to the spacing. For example, assuming that `chromosomes_units = 1000000` and the total ideogram size is 100 Mb,

default = 1u	1 Mb	break = 1u	1 Mb
default = 0.1r	10 Mb	break = 1u	1 Mb
default = 1u	1 Mb	break = 0.5r	0.5 Mb (50% of the spacing)
default = 0.1r	10 Mb	break = 0.25r	2.5 Mb (25% of the spacing)

Defining the break size to be relative is a good idea—it maintains constant the size of the break relative to spacing.

The appearance of the break is defined by the `axis_break_style`, which is set to a value for which a corresponding `<break_style>` block has been defined. For more details about formatting breaks, see the tutorial about spacing and breaks at

http://circos.ca/documentation/tutorials/ideograms/spacing_breaks/

```
<ideogram>
<spacing>

default = 0.1r
# when relative, size of break is computed
# relative to default spacing
break   = 0.25r

axis_break_at_edge = yes
axis_break        = yes
axis_break_style  = 2

<break_style 1>
stroke_color = vdgrey
```

```
# fills the break with a solid color, which has a
# thickness defined below
fill_color    = vdgrey
# thickness of the break fill, relative to
# ideogram thickness or absolute
thickness     = 0.25r
stroke_thickness = 2p
</break_style>

<break_style 2>
# when fill_color is not defined, the break space is empty, and
# the left & right edges of the space have a border
stroke_color   = vdgrey
stroke_thickness = 2p
# relative to ideogram thickness or absolute
thickness      = 2r
</break_style>
```

You can define multiple breaks

```
chromosomes_breaks = -chr3:13-17;-chr4:(-9;-chr4:41-);-chr5:11-19
```

The result of this break definition is shown in Figure 22. Note two things here. First, you can use “(“ and “)” to represent the start (or end) of the chromosome. Thus **0-9** is the same as **(-9**. This notation is helpful because you do not need to remember the start (or end) value of the chromosome to define a break at its edge.

Second, multiple breaks can be defined for a given chromosome. In this example, chromosome 4 has a break at its start **(-9**, which removes the first 9 Mb and at its end **41-**, which removes the last 9 Mb.

Lesson 8 – Ordering Ideogram Regions

The previous lesson showed how to define axis breaks, effectively splitting an ideogram into two or more pieces.

You may wonder whether it is possible to change the order of these pieces. Yes, and you use *tags* for this. A tag is an alternate name for an ideogram. Up to now, ideograms did not need to be explicitly named because each chromosome was associated with one ideogram. However, if you wish to split chromosome 4 into two ideograms and reorder them, these ideograms need to be given unique names. This is where *tags* are used.

Tags are delimited by [] and are defined in the `chromosomes` parameter. In the example below, chromosome 4 is associated with two ideograms. The first ideogram, named ‘a’, represents the region 0-11 Mb. The second ideogram, named ‘b’, represents 19-31 Mb.

```
chromosomes = chr3:0-11;chr4[4a]:0-11;chr4[4b]:19-31;chr5[5a]:9-21;chr5[5b]:29-41
```

Once ideograms of a chromosome are assigned tags, the tags can be used to reorder the ideograms using the `savme` method as we’ve seen previously (using the `chromosomes_order` parameter), except now instead of chromosome names, tags are used.

```
chromosomes_order = ^,chr1,4a,chr3,4b,-,5b,5a
```

You’ll recognize the order shown above. New is the `^` character, which represents the start of the ideogram circle. Ideogram 4a (`chr4:0-11`) is displayed after chromosome 1. Next comes chromosome 3, followed by ideogram 4b (`chr4:19-31`). The result is shown in Figure 23.

Ideograms formed by chromosomes that have been broken into pieces can be individually scaled by using the tag name of the ideogram.

```
chromosomes_scale = 5a=2,5b=2
```

The result is shown in Figure 24.

Lesson 9 – Cytogenetic Bands

For many genomes, the gross structure of chromosomes has been mapped using staining. These bands are called cytogenetic bands and are used as low-resolution navigational markers. Bands are optional in Circos and when used they serve to break up the ideogram segment into stripes.

The position of cytogenetic bands is defined in the karyotype file using band lines. You can use any color for the bands, but conventionally colors gposNNN (e.g., gpos25, gpos50, gpos100), gneg, gvar, acen and stalk are used. These are defined to match the conventional shading of cytogenetic bands in figures (see Figure 25). If you download the contents of a genome's karyotype table from the UCSC genome browser (*group Mapping and Sequencing Tracks, track Chromosome Band, table CytoBandIdeo*) you will see these color names in the last field of each line.

```
# the first field must be "band"
# the second field is the chromosome with which the band is associated
# next two fields are the name & label of the band (not used, but must be present)
# the last three fields are the start, end and color of the band.
band chr1 band1 band1 0 2500000 gneg
band chr1 band2 band2 2500000 5000000 gpos25
band chr2 band1 band1 0 2500000 gneg
band chr2 band2 band2 2500000 5000000 gpos25
band chr2 band3 band3 5000000 7500000 gpos100
band chr2 band4 band4 7500000 10000000 gvar
band chr3 band1 band1 0 1000000 stalk
band chr3 band2 band2 1000000 5000000 gpos50
band chr3 band3 band3 5000000 7500000 gpos100
band chr3 band4 band4 7500000 10000000 gvar
band chr3 band5 band5 10000000 15000000 acen
band chr3 band7 band7 15000000 19000000 gneg
...
...
```

Figure 26 shows how the bands appear as defined for chromosomes 1, 2 and 3. You do not need to define bands for all of the chromosomes.

TRANSPARENT COLORS

The transparency value should be 1...N where N is defined by `auto_alpha_steps` in the `<image>` block. The `auto_alpha_steps` parameter defines the number of transparent colors that are automatically allocated for each color definition (e.g. such as those defined by default in `etc/color.conf`). A transparent color has the name `COLOR_an` where n defines the transparency level (1 – least transparent, N – most transparent, n=1-N) and COLOR is the original name of the color. For example, `red_a1` is a nearly opaque version of red and `red_a5` is an almost transparent red.

You can adjust the transparency of the band pattern by using `band_transparency` within the `<ideogram>` block. Figure 27 shows the effect of setting `band_transparency=3`, where each band color was interpreted as `COLOR_a3`.

Lesson 10 – Drawing Multiple Genomes

Thus far, we've drawn figures of a sample genome (see `2/data/karyotype.5chr.txt`) and its banded equivalent (`2/data/karyotype.5chr.banded.txt`). Karyotypes of actual genomes (human, chimp, mouse, rat, etc) can be created from data available through the UCSC genome browser table viewer (<http://genome.ucsc.edu/cgi-bin/hgTables>).

Karyotype information about a genome can be obtained by selecting **group Mapping and Sequencing Tracks** and **track Chromosome Band or Chromosome Band (Ideogram)**. For genomes that have not been assembled into entire chromosomes, this track is not available.

Karyotype files for human and mouse are available in `data/karyotype` in the Circos distribution directory.

`data/karyotype/karyotype.human.hg19.txt` – human karyotype (GRCh37/hg19)

`data/karyotype/karyotype.mouse.mm9.txt` – mouse karyotype (NCBI37/mm9)

You'll also find symbolic links `karyotype.human.txt` and `karyotype.mouse.txt` pointing to the karyotypes of the newest assembly. Karyotypes of other common species are also provided (rat, chimp, fly, etc.).

In examples above, chromosome names were `chrN` (e.g. `chr1`, `chr2`, ...). Because chromosome names must be unique, to draw multiple genomes in the same figure it is convenient to name chromosomes after the corresponding species. Thus, human (*Homo sapiens*) chromosomes are prefixed with "hs" (`hs1`, `hs2`, ...), mouse (*Mus musculus*) are prefixed with "mm" (`mm1`, `mm2`, ...).

```
# data/karyotype/karyotype.human.txt
chr - hsX hsX 0 155270560 chrx
chr - hsY hsy 0 59373566 chry
chr - hs1 hs1 0 249250621 chr1
chr - hs2 hs2 0 243199373 chr2
chr - hs3 hs3 0 198022430 chr3
chr - hs4 hs4 0 191154276 chr4
chr - hs5 hs5 0 180915260 chr5
chr - hs6 hs6 0 171115067 chr6
chr - hs7 hs7 0 159138663 chr7
chr - hs8 hs8 0 146364022 chr8
chr - hs9 hs9 0 141213431 chr9
chr - hs10 hs10 0 135534747 chr10
chr - hs11 hs11 0 135006516 chr11
chr - hs12 hs12 0 133851895 chr12
chr - hs13 hs13 0 115169878 chr13
chr - hs14 hs14 0 107349540 chr14
chr - hs15 hs15 0 102531392 chr15
chr - hs16 hs16 0 90354753 chr16
chr - hs17 hs17 0 81195210 chr17
chr - hs18 hs18 0 78077248 chr18
chr - hs19 hs19 0 59128983 chr19
chr - hs20 hs20 0 63025520 chr20
chr - hs21 hs21 0 48129895 chr21
chr - hs22 hs22 0 51304566 chr22
```

By setting the `karyotype` parameter to point to the human karyotype file

```
karyotype = ../../data/karyotype/karyotype.human.txt
#karyotype = ../../data/karyotype/karyotype.mouse.txt
```

we obtain a figure of all the human chromosomes (Figure 28). When the `karyotype.mouse.txt` file is used, we have the mouse genome, shown in Figure 29.

CHROMOSOME COLOR SCHEME

The UCSC genome browser uses a color convention to index human chromosomes. These colors are shown in Figure 30 and are defined in the color definition file `etc/color.ucsc.conf` in the Circos distribution. Chromosome M is the mitochondrial chromosome and chromosome Un (`chrUn`) is a concatenation of all unanchored sequence (sequence that has not yet been, or cannot be, associated with a specific chromosome).

```
# UCSC genome browser RGB colors for human chromosomes
chr1 = 153,102,0
chr2 = 102,102,0
chr3 = 153,153,30
chr4 = 204,0,0
chr5 = 255,0,0
chr6 = 255,0,204
chr7 = 255,204,204
chr8 = 255,153,0
chr9 = 255,204,0
chr10 = 255,255,0
chr11 = 204,255,0
chr12 = 0,255,0
chr13 = 53,128,0
chr14 = 0,0,204
chr15 = 102,153,255
chr16 = 153,204,255
chr17 = 0,255,255
chr18 = 204,255,255
chr19 = 153,0,204
chr20 = 204,51,255
chr21 = 204,153,255
chr22 = 102,102,102
chr23 = 153,153,153
chrX = 153,153,153
chr24 = 204,204,204
chrY = 204,204,204
chrM = 204,204,153
chrO = 204,204,153
chrUn = 121,204,61
chrNA = 255,255,255
```

The conventional human chromosome colors are named after the chromosome number and prefixed with 'chr'. Thus, chromosome 1, which is named `hs1` is associated with color `chr1`. These colors are listed above and shown in Figure 30. Each of these colors has a synonym that starts with `hs` (i.e. `chr5` and `hs5` are both the same color).

These chromosome colors are used extensively by the conservation tracks in the UCSC genome browser, and you should use them to create a color key for Circos ideograms if you want to preserve this color convention. In the next session, you will learn about perceptual qualities of color and about the differences of how the color is specified (e.g. RGB, HLS) and how it is perceived.

LUMINANCE NORMALIZED COLORS

A quick example of this is shown in Figure 31, where the UCSC color palette is shown with its luminance-normalize counterparts. Notice that the yellow used for chromosome 10 appears much brighter than other colors (highest luminance), drawing attention to this chromosome. By normalizing the colors to have the same perceived brightness (inner rings in Figure 31), the color scheme becomes visually more uniform. Luminance normalized colors are available for each chromosome color. These have prefixes `lum90`, `lum80` and `lum70` (e.g. `lum80chr11`).

In Circos, chromosome color assignment is done in the karyotype file—the color is the last field in the file. Recall that the sample 5-chromosome genome karyotype file was defined as

```
chr - chr1 1 0 5000000 spectral-5-div-1
chr - chr2 2 0 10000000 spectral-5-div-2
chr - chr3 3 0 20000000 spectral-5-div-3
chr - chr4 4 0 50000000 spectral-5-div-4
chr - chr5 5 0 100000000 spectral-5-div-5
```

where `spectral-5-div-N` corresponded to colors from a 5-color diverging spectral Brewer palette (Figure 1).

```
spectral-5-div-1 = 215,25,28
spectral-5-div-2 = 253,174,97
spectral-5-div-3 = 255,255,191
spectral-5-div-4 = 171,221,164
spectral-5-div-5 = 43,131,186
```

If you wish to display chromosomes from multiple genomes, specify multiple files as a comma-separated list for the `karyotype` parameter. Keep in mind that all chromosome names must be unique.

```
karyotype = data/karyotype.human.txt,data/karyotype.mouse.txt
```

Now that you're drawing chromosomes from different species, it's good to have labels reflect this. In the karyotypes the labels of the chromosomes do not contain the species prefix. You can change the label string using `label_format` in `ideogram.conf`.

```
label_format = eval(sprintf("%s",var(chr)))
```

This will change the label to be the same as the chromosome name, which can be referenced with `var(chr)`. We'll see the `var()` syntax in other examples.

The human and mouse chromosomes are drawn in Figure 32. The tick labels are starting to overlap, which we will fix later.

Lesson 11 – Ideogram, Color, Scale, Order and Orientation

In this lesson we will develop the ideogram layout used for the next session, in which we will discuss data tracks.

In Figure 32, a total of 45 ideograms are displayed (24 human and 21 mouse chromosomes). Being able to quickly and easily control which ideograms are displayed and adjusting their format is important. In this lesson, I'll show you how to manipulate what ideograms are displayed, their order, color and orientation.

FILTERING IDEOGRAMS

If you wish to remove several ideograms from the display (e.g. hs1, hs2, mm1, mm2) specify their names in the `chromosomes` parameter, with a “ - ” prefix (Figure 33).

```
chromosomes_display_default = yes
chromosomes = -hs1;-hs2;-mm1;-mm2
```

You can remove ideograms using a regular expression. For example, to remove (hsX, hsY, mmX, mmY), use a regular expression instead of an ideogram name. Regular expressions are delimited with / /.

```
chromosomes_display_default = yes
chromosomes = -/[XY]/
```

You can combine ideogram names with regular expressions to form a complex filter. In the example below, removed are (hs1, mm1, hsY, mmY) and all chromosomes with two digits in their name (hs10...hs22, mm10...mm19).

```
chromosomes_display_default = yes
chromosomes = -hs1;-mm1;-/Y/;-\d\d/
```

Always use “;” as the delimiter between records in the `chromosomes` parameter—“,” is not a valid delimiter because it is reserved for a separator in the specifying regions (e.g. 0-10, 15-20).

In the above examples, the desire was to keep all the chromosomes defined in the karyotype file (`chromosomes_display_default=yes`) but remove one or more from the display. In the converse, you can define which ideograms to display by setting `chromosomes_display_default=no` and building the ideogram list using the `chromosomes` parameter. Both ideogram names and regular expressions are supported. For example, to display hs1..hs5, hs10, and mm10-mm15 and mmX, as shown in Figure 34, use

```
chromosomes_display_default = no
chromosomes = /hs[1-5]$;/hs10;/mm1[0-5]/;mmX
```

Regular expressions are not case-insensitive but can match a substring of the ideogram name. Thus /hs[1-5]/ will match not only hs1..hs5, but also hs10, hs11, hs12, etc. Use anchors (^ for start of string and \$ for end of string) to limit the match. Thus /hs[1-5]\$/ will only match hs1...hs5.

IDEOGRAM COLOR

The color of each chromosomes, and therefore all of its ideograms, is specified in the karyotype file. By convention, the colors of human chromosomes are set to the UCSC color convention (Figure 30) and all other species' chromosomes are set to white.

To override this, use `chromosomes_color`, which accepts both regular expressions and ideogram names. Figure 35 shows the effect of changing human chromosomes to red and mouse ones to blue.

```
chromosomes_display_default = yes
chromosomes_color          = /hs/:reds-5-seq-5,/mm/:blues-5-seq-5
```

Both “=” and “:” can be used as the assignment operator (e.g. /hs/:reds-5-seq-5 and /hs/=reds-5-seq-5). For parameters other than chromosomes and chromosomes_breaks, “,” can also be used to delimit records.

IDEOGRAM PROGRESSION

The orientation of ideograms can be reversed using `chromosomes_reverse`, which accepts both ideogram names and regular expressions, just like `chromosomes_color`. Thus, to reverse all mouse chromosomes

```
chromosomes_reverse = /mm/
```

RELATIVE IDEOGRAM SCALING

Recall that we've already used `chromosomes_scale` to adjust the magnification of an ideogram (e.g. `chromosomes_scale = chr1=0.5`). When the scale was relative, the ideogram was made to occupy a fraction of the image (e.g. `chr1=0.5r`).

When a large number of ideograms is shown, you can normalize the length of ideograms by using a regular expression in the `chromosomes_scale` parameter. For example, to show only human chromosomes and make each to have the same length in the image

```
chromosomes_display_default = no
chromosomes                 = /hs/
```

```
chromosomes_scale = /hs/=1rn
```

The scale value is derived from the fact that there are 24 human chromosomes, and to make each one occupy the same fraction (i.e. $1/24^{\text{th}}$ of the image), a relative scale of $1/24 = 0.0417$ is required (e.g. `0.0417rn`) By using the “`rn`” suffix, we can calculate this scale automatically. Figure 36 shows the result.

The figure can be partitioned evenly into ideogram sets by assigning a different scale to groups of ideograms. Figure 37 shows the result of the block below, which divides the image in half: 50% occupied by 5 human chromosomes and 50% by 3 mouse chromosomes. Again, by using “`rn`” we don’t need to remember how many of each type of chromosome are shown.

```
chromosomes_display_default = no
chromosomes = /hs[1-5]$;/mm1[7-9]/
chromosomes_scale = /hs/:0.5rn,/mm/:0.5rn
```

SYMMETRIC IDEOGRAM LAYOUT

You now have the knowledge to create the symmetric layout shown in Figure 38, in which 2 human and 2 mouse chromosomes are shown, each occupying 25% of the image.

```
chromosomes_display_default = no
chromosomes = hs1;hs2;mm1;mm2
chromosomes_order = hs1,hs2,mm2,mm1
chromosomes_color = hs1:rdylbu-11-div-2,hs2:rdylbu-11-div-3,
                    mm1:rdylbu-11-div-10,mm2:rdylbu-11-div-9
chromosomes_reverse = /mm/
chromosomes_scale = ./:1rn
```

The regular expression `/./` will match everything and is a useful catch-all for ideograms.

Lesson 12 – Relative and Absolute Ticks

In this final lesson of this session, we'll refine Figure 38 and prepare it for addition of data tracks.

Notice that in `etc/ideogram.conf` for this lesson, there are several parameters that end in `"*"`. This suffix indicates that the corresponding parameter values should override any previously seen definitions of this parameter. In this case, the included file `../etc/ideogram.conf` defines `label_radius`, `band_stroke_thickness`, `band_stroke_color` and `band_transparency`. To customize the image in this lesson, we don't want to edit this file, because that would change all the other images in this session that import it. Instead, we override the parameters. The `"*"` is required because the same parameter cannot be defined twice.

```
<<include ../../etc/ideogram.conf>>

label_radius* = dims(ideogram, radius) + 0.3r

band_stroke_thickness* = 1
band_stroke_color*     = black
band_transparency*     = 5
```

You can employ multiple levels of overriding parameters, by using more than one `"*"` in the suffix. Thus, `label_radius` will be overridden by `label_radius*`, which will be overridden by `label_radius**`, and so on.

In Figure 39 you will notice that there are two tick mark rings. The inner tick mark ring is relative (0%–100%) and the outer is absolute. The position, spacing, size and labels of ticks is defined in the `<ticks>` block. This block can become lengthy, so it is convenient to relegate it to a separate file which is imported into the central configuration file.

```
# 2/12/etc/ticks.conf
show_ticks      = yes
show_tick_labels = yes
show_grid       = yes

<ticks>
tick_label_font = light
radius          = dims(ideogram, radius_outer) + 45p
label_offset    = 5p
label_size      = 8p
multiplier      = 1e-6
color           = black

# absolute tick, every 20 Mb, with label and grid

<tick>
spacing        = 20u
size           = 12p
thickness      = 2p
show_label     = yes
label_size     = 10p
```

```
format          = %d

grid_start      = 1r
grid_end        = 1r+45p
grid_color      = vdgrey
grid_thickness   = 1p
grid            = yes
</tick>

# absolute tick, every 10 Mb, with grid
<tick>
spacing         = 10u
size            = 7p
thickness       = 2p
show_label      = no

grid_start      = 1r
grid_end        = 1r+45p
grid_color      = grey
grid_thickness   = 1p
grid            = yes
</tick>

# absolute tick, every 2 Mb
<tick>
spacing         = 2u
size            = 3p
thickness       = 2p
show_label      = no
</tick>

# relative tick, every 2%
<tick>
radius          = 0.75r
spacing_type    = relative
rspacing        = 0.02
size            = 3p
thickness       = 1p
show_label      = no
</tick>

# relative tick, every 10%, with label and grid
<tick>
radius          = 0.75r
spacing_type    = relative
rspacing        = 0.10
size            = 6p
thickness       = 1p
show_label      = yes
label_relative  = yes
rmultiplier     = 100
format          = %d
suffix          = %

grid_start      = 0.5r
grid_end        = 0.75r
grid_color      = grey
```

```
grid_thickness = 1p
grid          = yes

</tick>

</ticks>
```

Although the tick definition looks complicated, it can be broken down into relatively simple elements. Ticks are divided into groups, with each group defined in a separate `<tick>` block. All ticks in a group have the same spacing, and multiple groups allow to create independently formatted ticks for different spacings. In this example, there are absolute tick groups for spacing of 20, 10 and 2 Mb. First the 20 Mb ticks are drawn (0, 20, 40, 60 Mb), then the 10 Mb ticks are drawn, but *only in places where no other ticks have already been placed*. Thus the 10 Mb ticks are drawn at 10Mb, 30Mb, 50Mb, and so on (they are not drawn at 20, 40 Mb, ... because the 20 Mb ticks already occupy those positions). Finally, the 2 Mb ticks are drawn (2, 4, 6, 8, 12 Mb, ...).

Each tick group can have its own grid, formed by radial lines from `grid_start` radial position to `grid_end` radial position. These positions are typically specified in relative units (relative to ideogram circle). A version of Figure 39 with grids is shown in Figure 40.

As one last element, we'll add a color key to the ideograms. This will be achieved by using a `<highlights>` block. This is the first block thus far which requires a data file. The data file defines genome regions (which may overlap) for which a colored segment is drawn in the figure. The highlight segments are drawn underneath any ticks and grids and are useful for annotating parts of the figure with colors (for lookup or focus).

The highlights will be defined as follows

```
<highlights>
<highlight>
file = ../data/highlight.txt
r0   = 1r+40p
r1   = 1r+45p
</highlight>
</highlights>
```

All regions found in the file `2/data/highlight.txt` will be drawn as segments between radial positions `r0 = 1r+40p` and `r1 = 1r+45p`. This radial position format expands on expressions you've already seen (e.g. `1.125r`) by adding an absolute value offset. The expression `1r+40p` means a radial position obtained by adding 40 pixels to the outer rim of the ideogram circle. Similarly, `1r+45p` means a position obtained by adding 45 pixels.

You can have any number of `<highlight>` blocks. The blocks may refer to highlight regions that overlap both radially and with respect to their genomic coordinates.

The contents of the `2/data/highlight.txt` file define the highlight regions.

```
# circos/workstation/2/data/highlight.txt
# chr start end options
hs1 0 249250621 fill_color=rdylbu-11-div-2
```

```
hs2 0 243199373 fill_color=rdylbu-11-div-3
mm1 0 197195432 fill_color=rdylbu-11-div-10
mm2 0 181748087 fill_color=rdylbu-11-div-9
```

Each chromosome has one region, which spans the full extent of the chromosome. The option parameter is used to set formatting options for the highlight regions. In this case, the `fill_color` is to correspond to the ideogram color.

The final figure, with highlights, is shown in Figure 41.

Figures

Figure 1	32
Figure 2	33
Figure 3	34
Figure 4	35
Figure 5	36
Figure 6	37
Figure 7	38
Figure 8	39
Figure 9	40
Figure 10	41
Figure 11	42
Figure 12	43
Figure 13	44
Figure 14	45
Figure 15	46
Figure 16	47
Figure 17	48
Figure 18	49
Figure 19	50
Figure 20	51
Figure 21	52
Figure 22	53
Figure 23	54
Figure 24	55
Figure 25	56
Figure 26	57
Figure 27	58

Figure 28	59
Figure 29	60
Figure 30	61
Figure 31	62
Figure 32	63
Figure 33	64
Figure 34	65
Figure 35	66
Figure 36	67
Figure 37	68
Figure 38	69
Figure 39	70
Figure 40	71
Figure 41	72

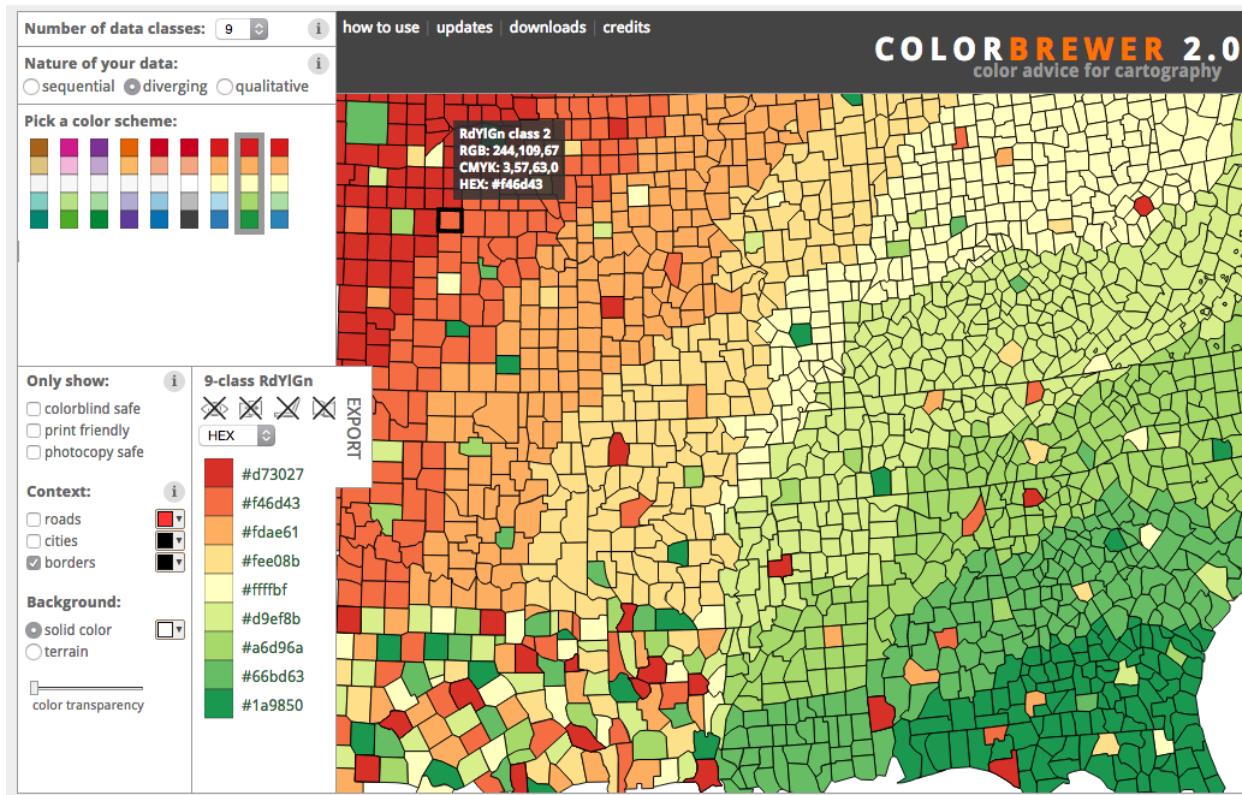


FIGURE 1

A 9-color spectral Brewer palette, as shown with Colorbrewer 2.0 (retrieved 27 Apr 2016).

For more palettes, see www.colorbrewer.org.

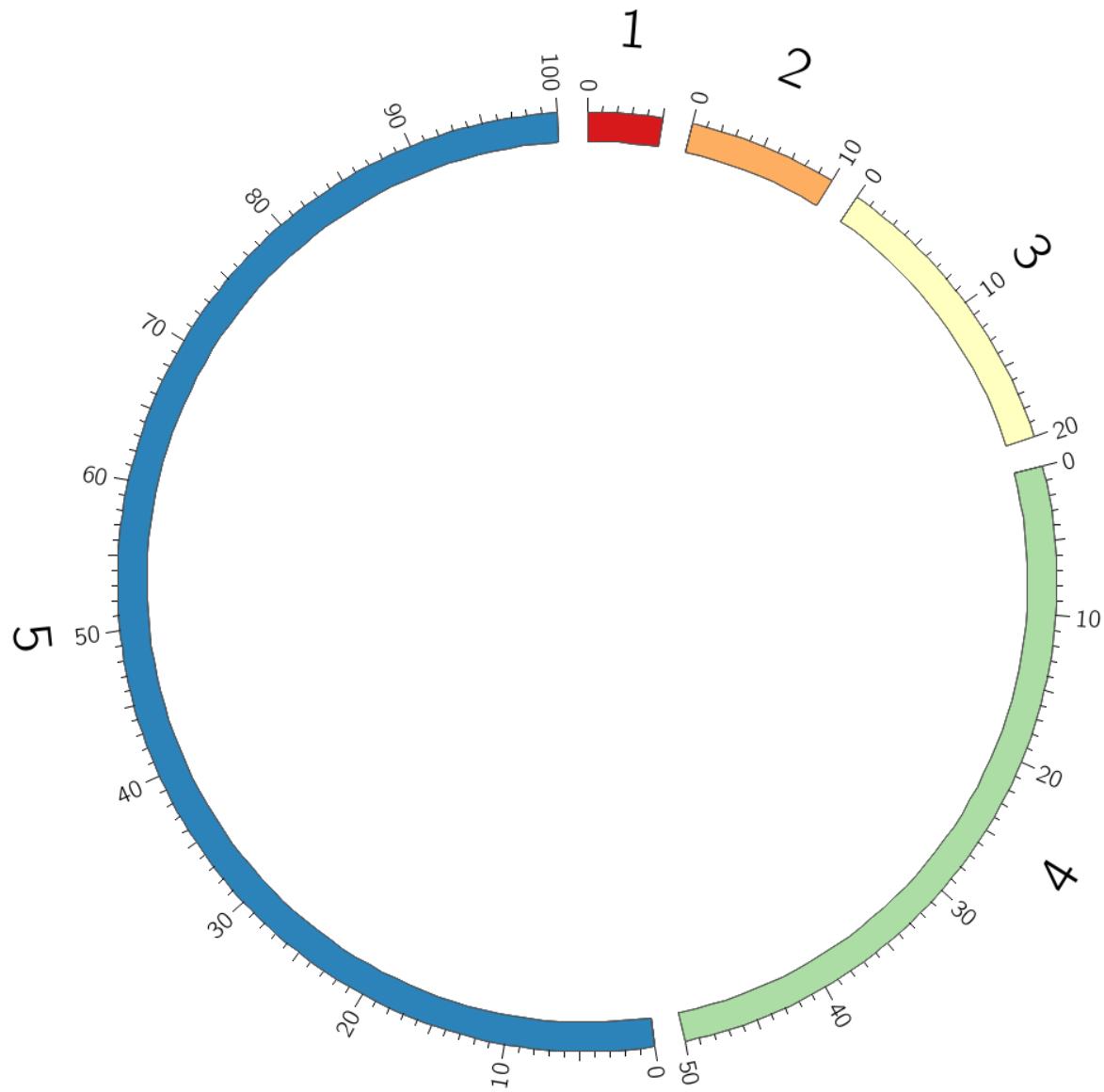


FIGURE 2

Five ideograms, arranged clockwise from 12 o'clock.

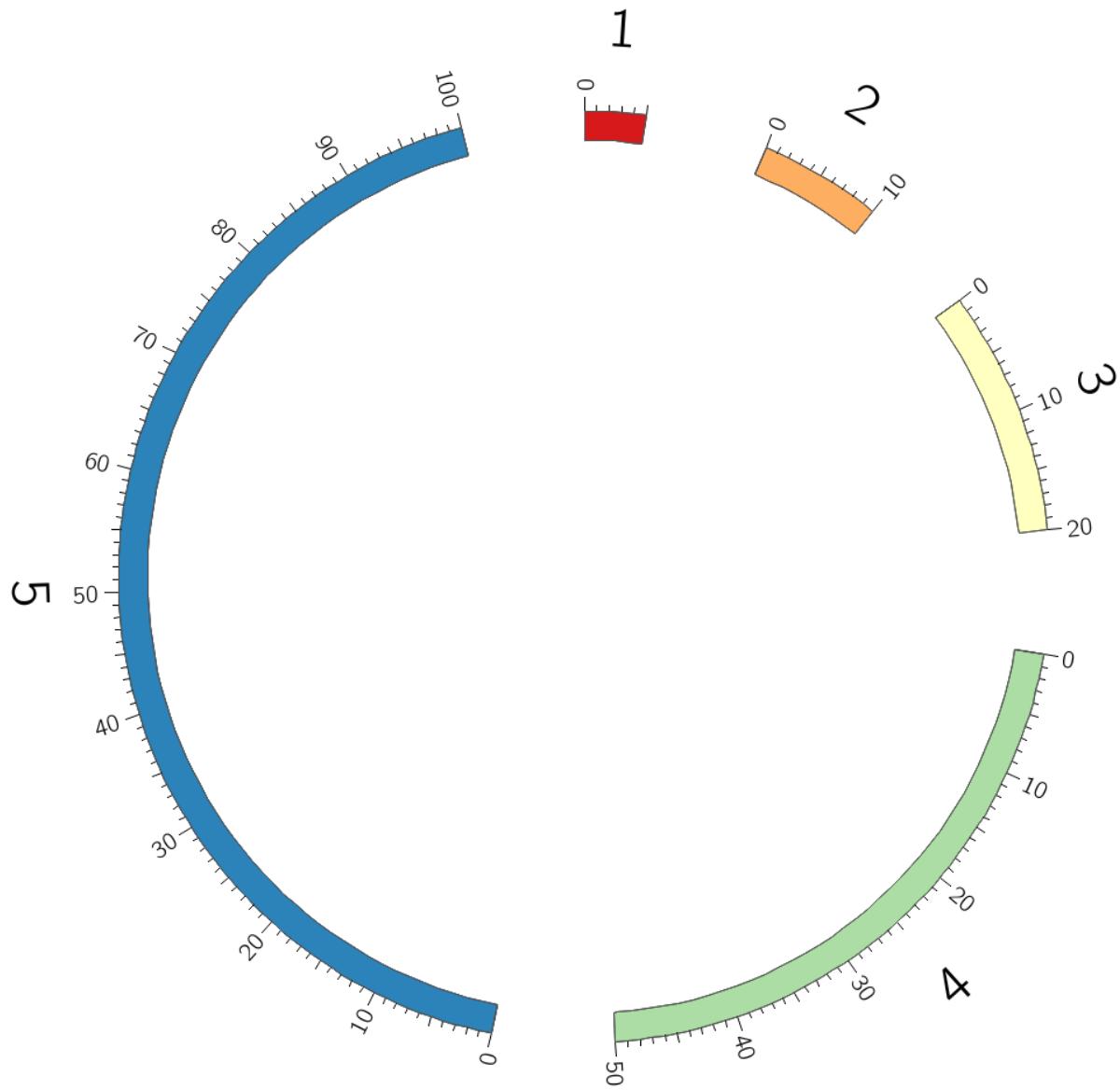


FIGURE 3

Changing the spacing from 2 μ to 10 μ has increased spacing by 5-fold.

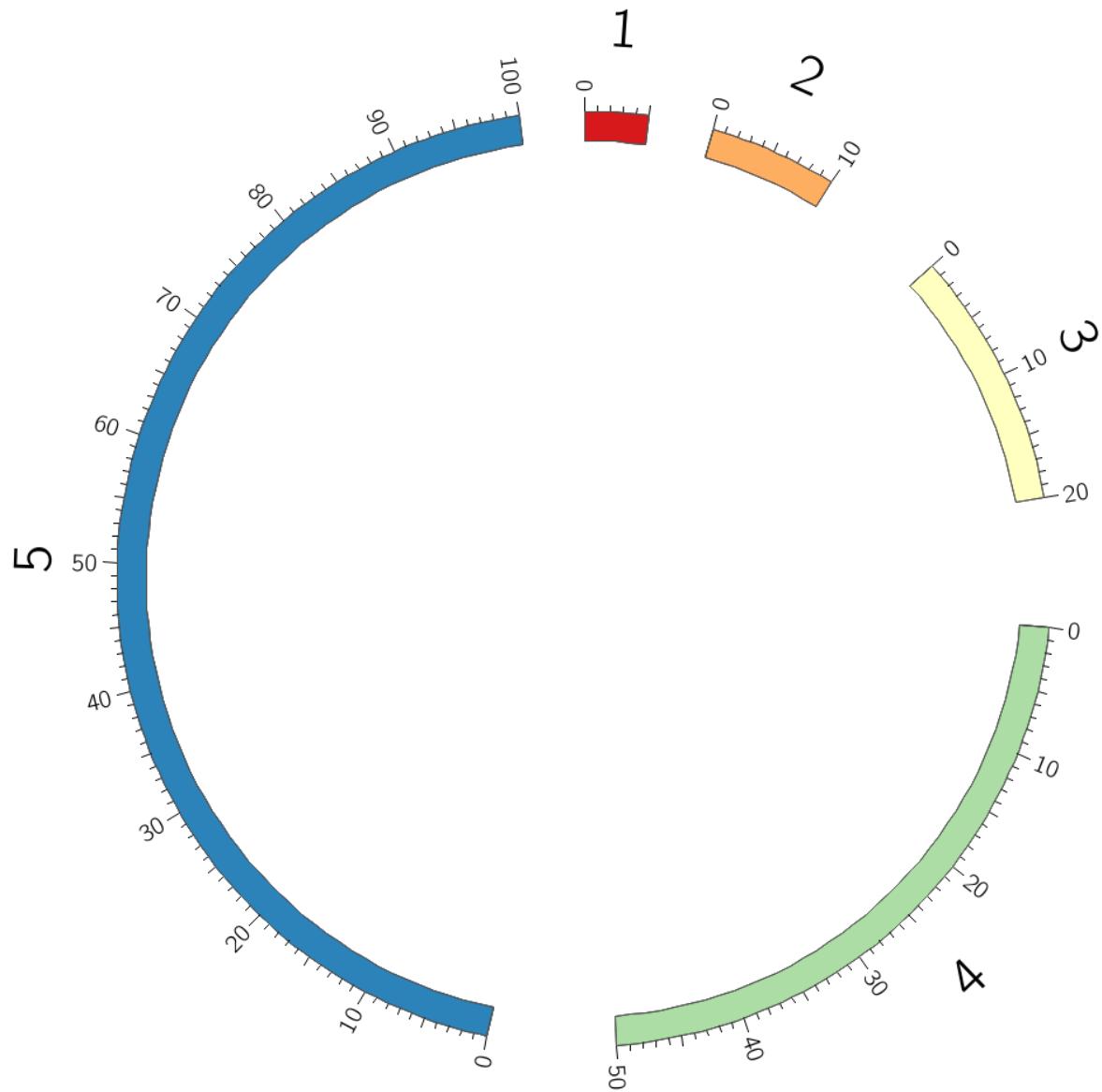


FIGURE 4

Spacing between individual ideogram pairs can be adjusted independently. This is done using the `<pairwise>` block.

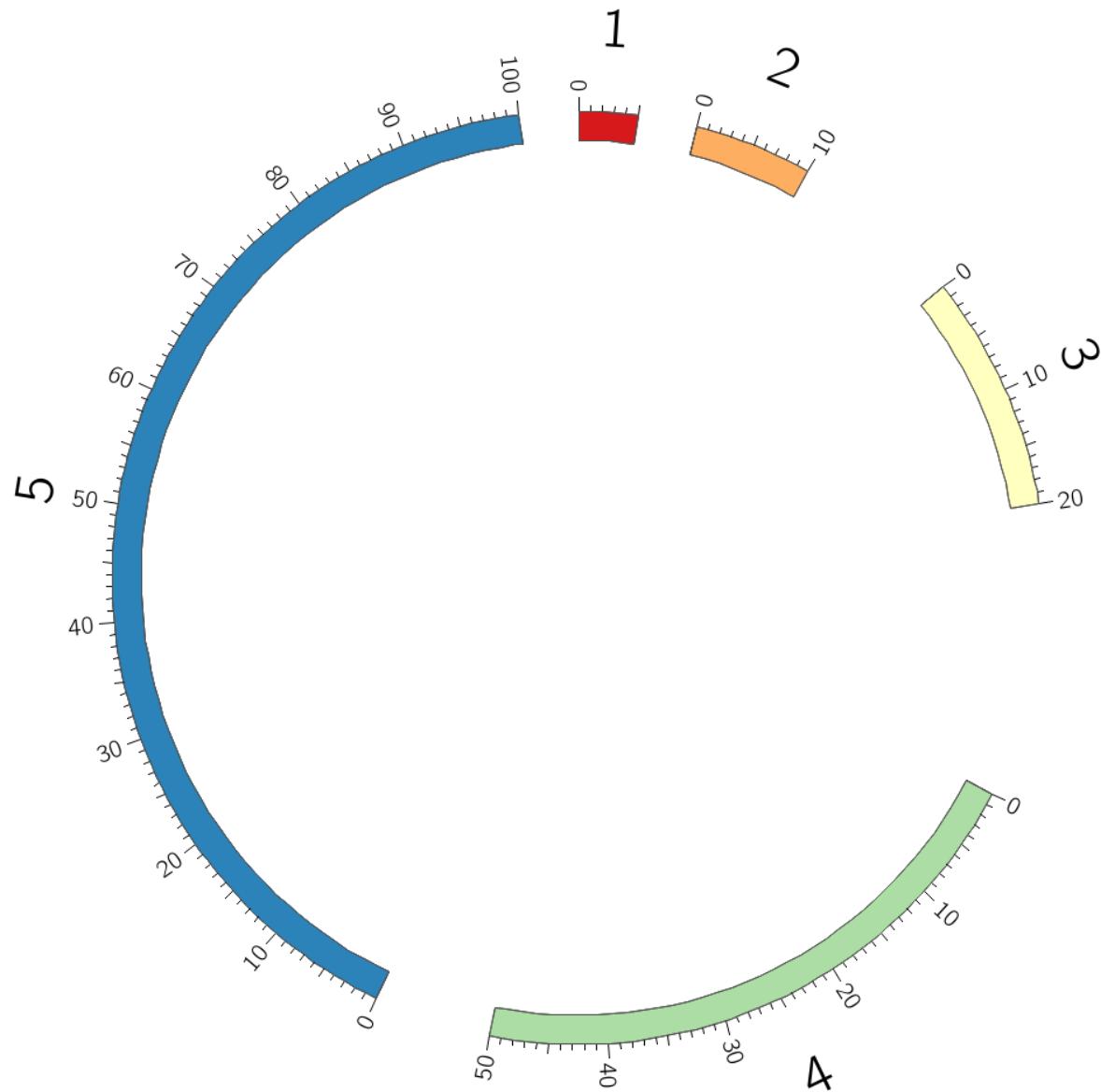


FIGURE 5

Spacing between individual ideogram pairs can be adjusted independently. This is done using the `<pairwise>` block. In addition to a change in spacing around `chr1`, spacing between `chr2` and `chr3` is changed to `15u` and between `chr3` and `chr4` to `25u`.

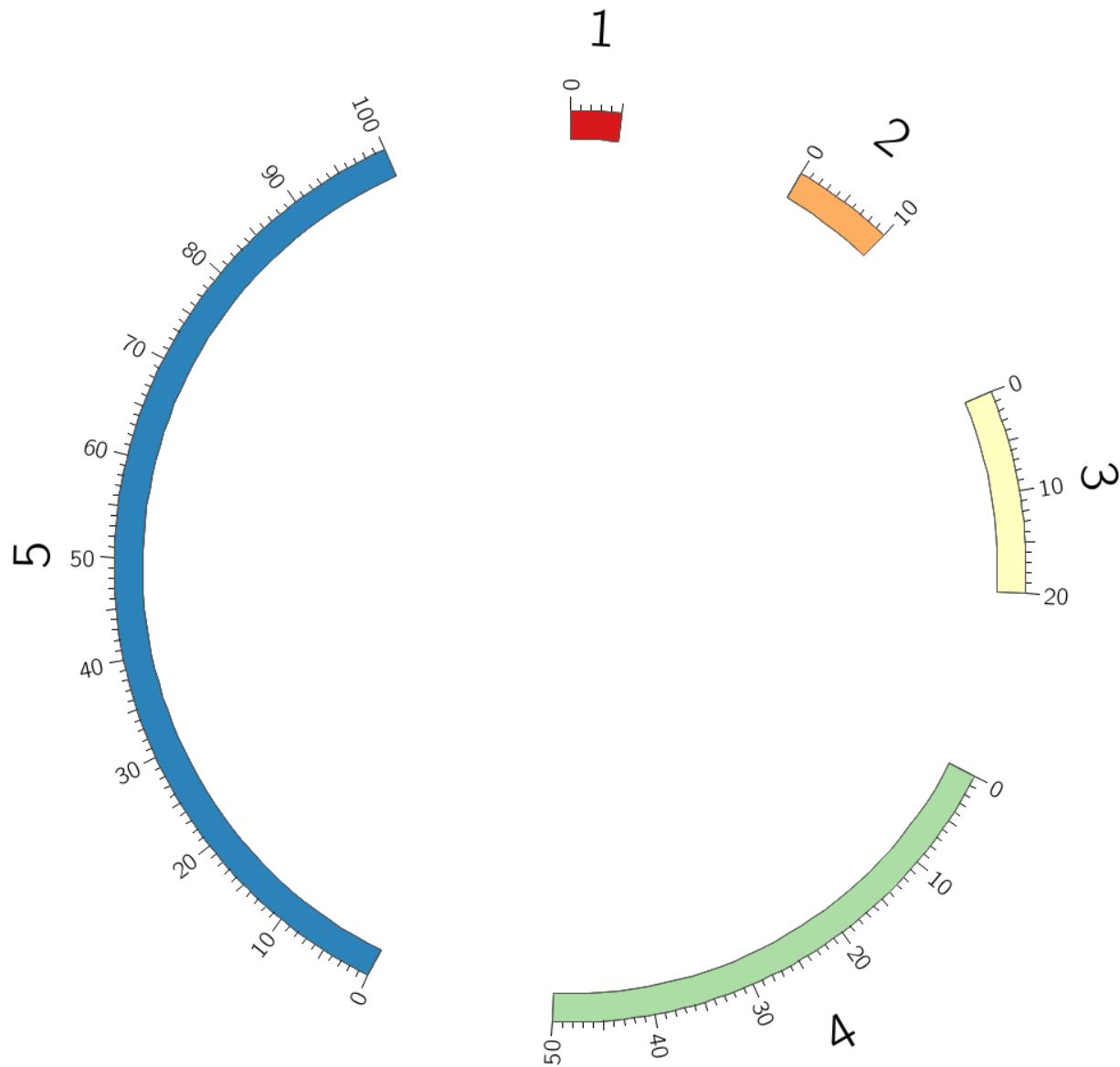


FIGURE 6

Ideogram spacing can be made relative to total ideogram size by using the `r` suffix. For example, the block below sets each space to 10% of total ideogram size.

```
<spacing>
default = 0.1r
...
</spacing>
```

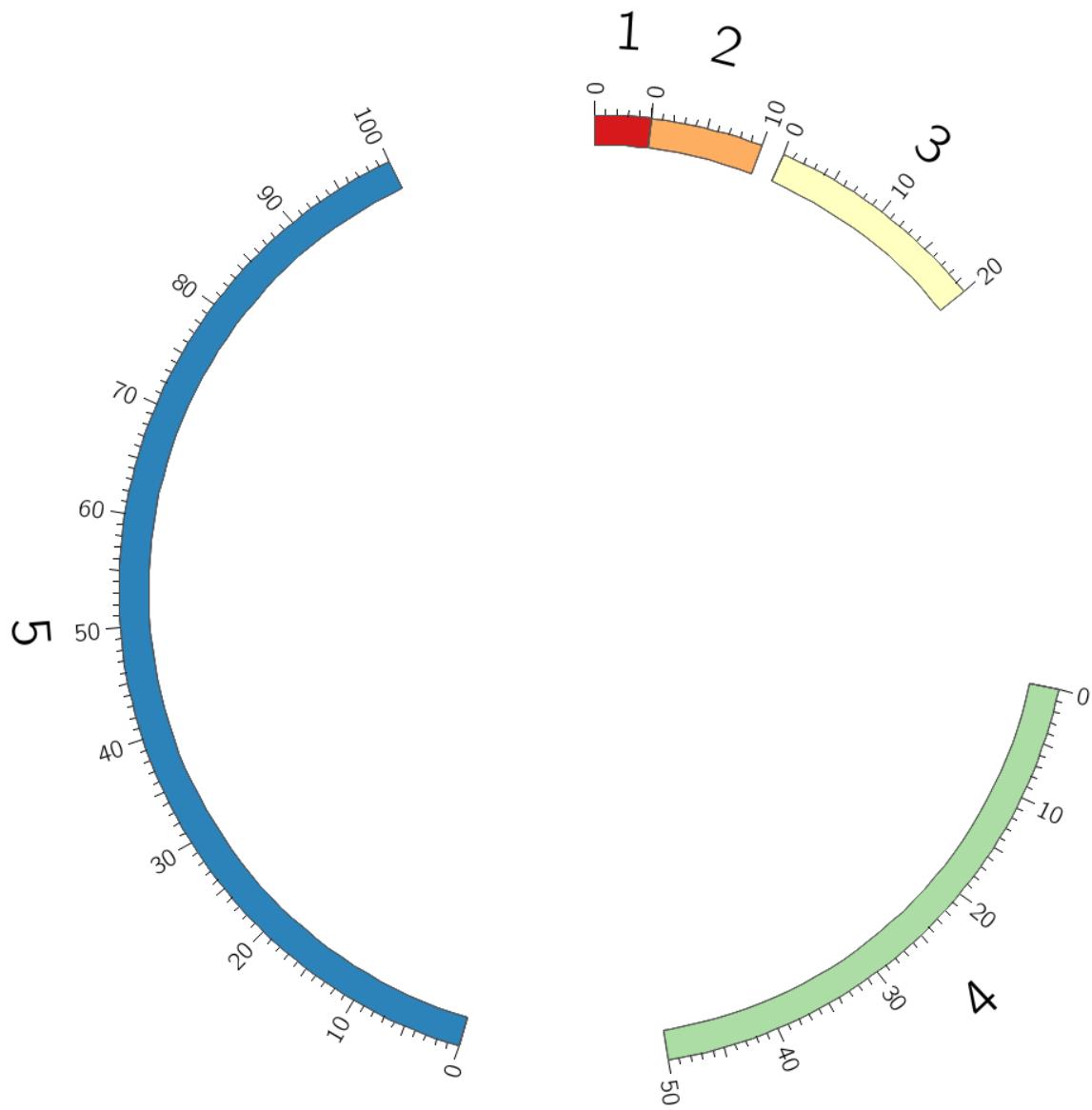


FIGURE 7

Absolute and relative spacing can be used in the same figure. Here, default ideogram spacing is relative (0.1r), but spacing between the first four chromosomes is set to 0u, 2u and 2r, respectively.

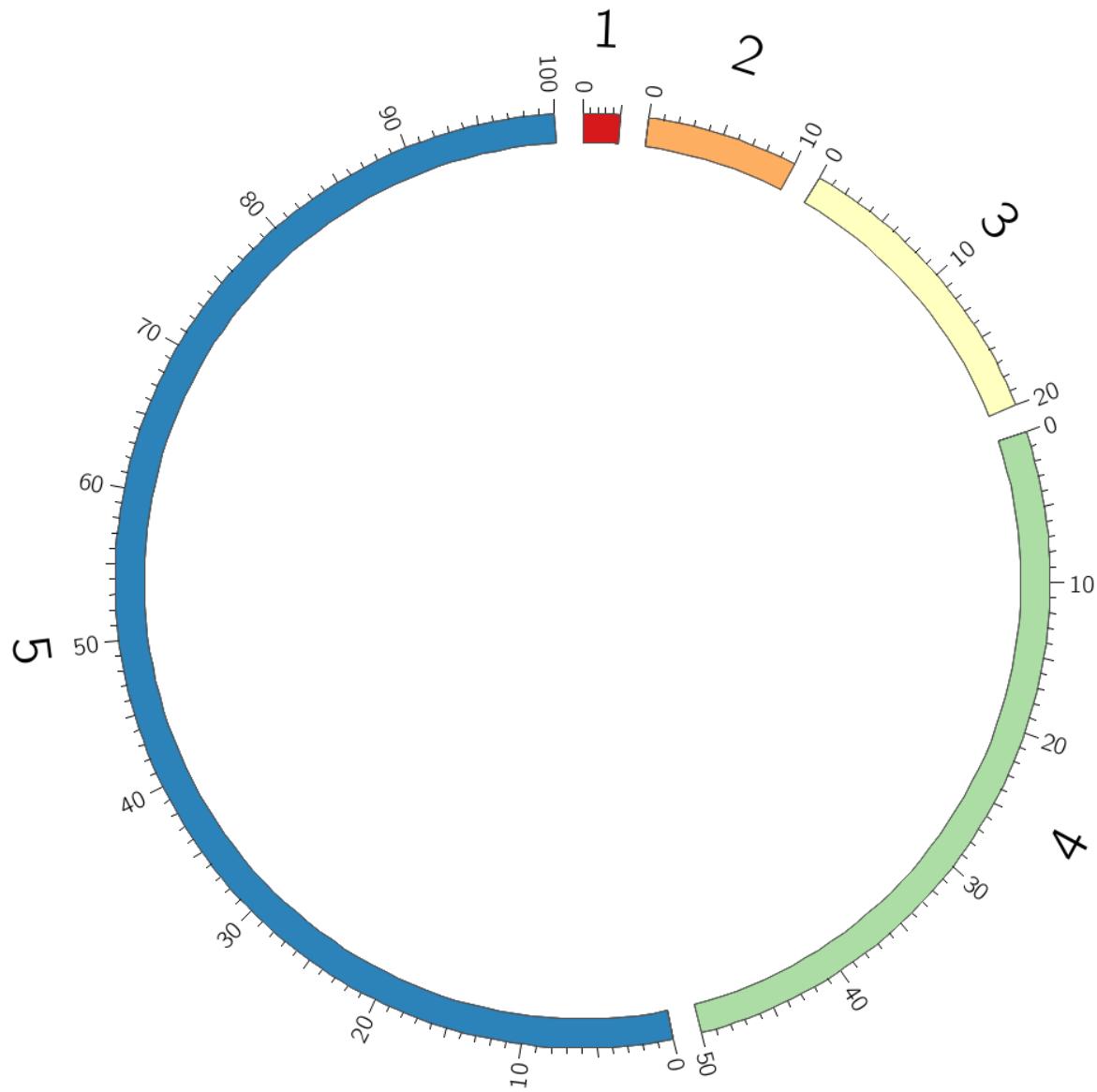


FIGURE 8

By changing the scale of chromosome 1 to 0.5 using the `chromosomes_scale` parameter, the ideogram of this chromosome is shrunk by 2x.

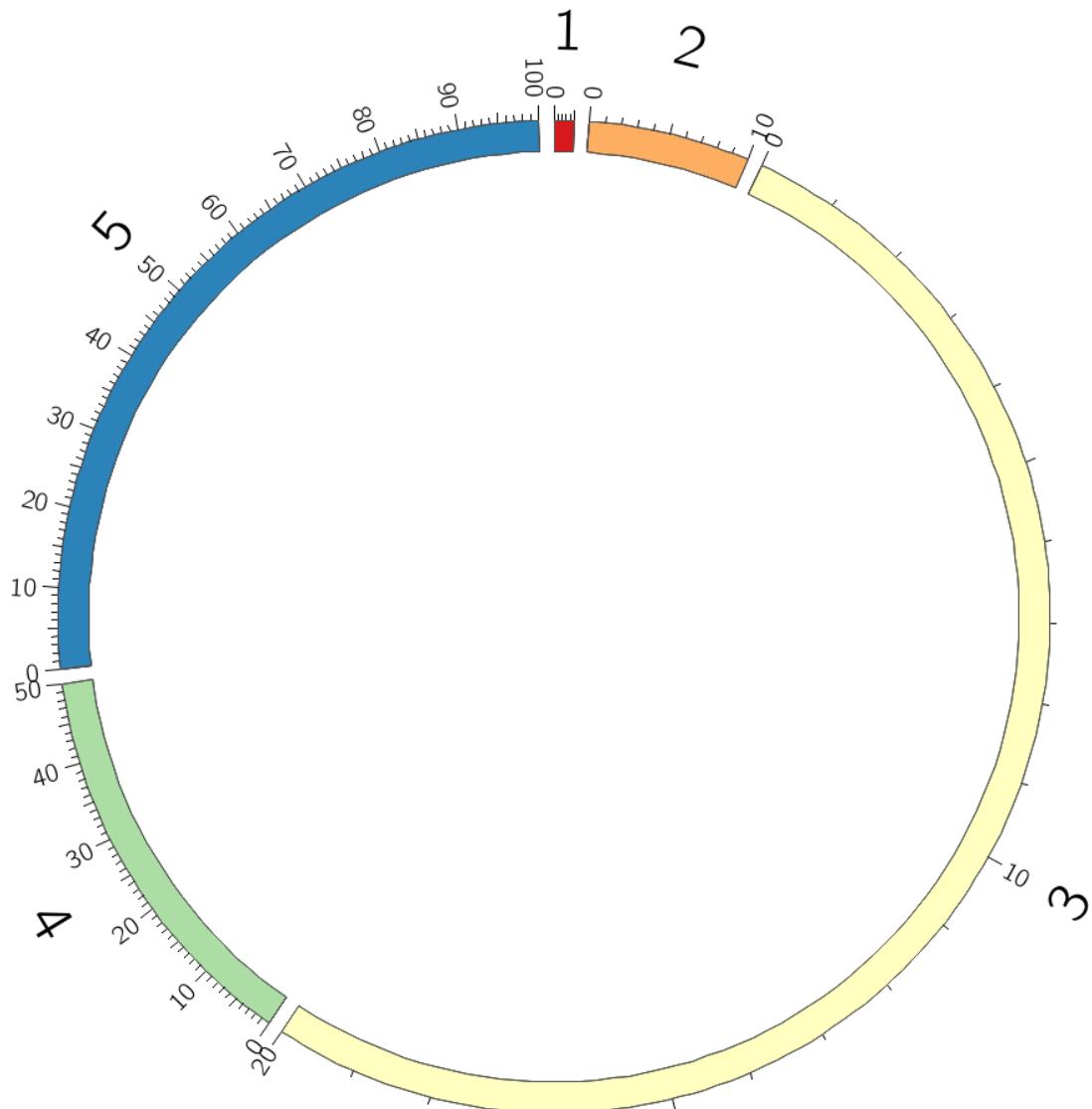


FIGURE 9

By changing the scale of chromosome 1 to 0.5, chromosome 2 to 2x and chromosome 3 to 10x, the figure's focus shifts to chromosome 3 (which is now magnified 10x) and away from chromosome 1 (which is reduced 2x).

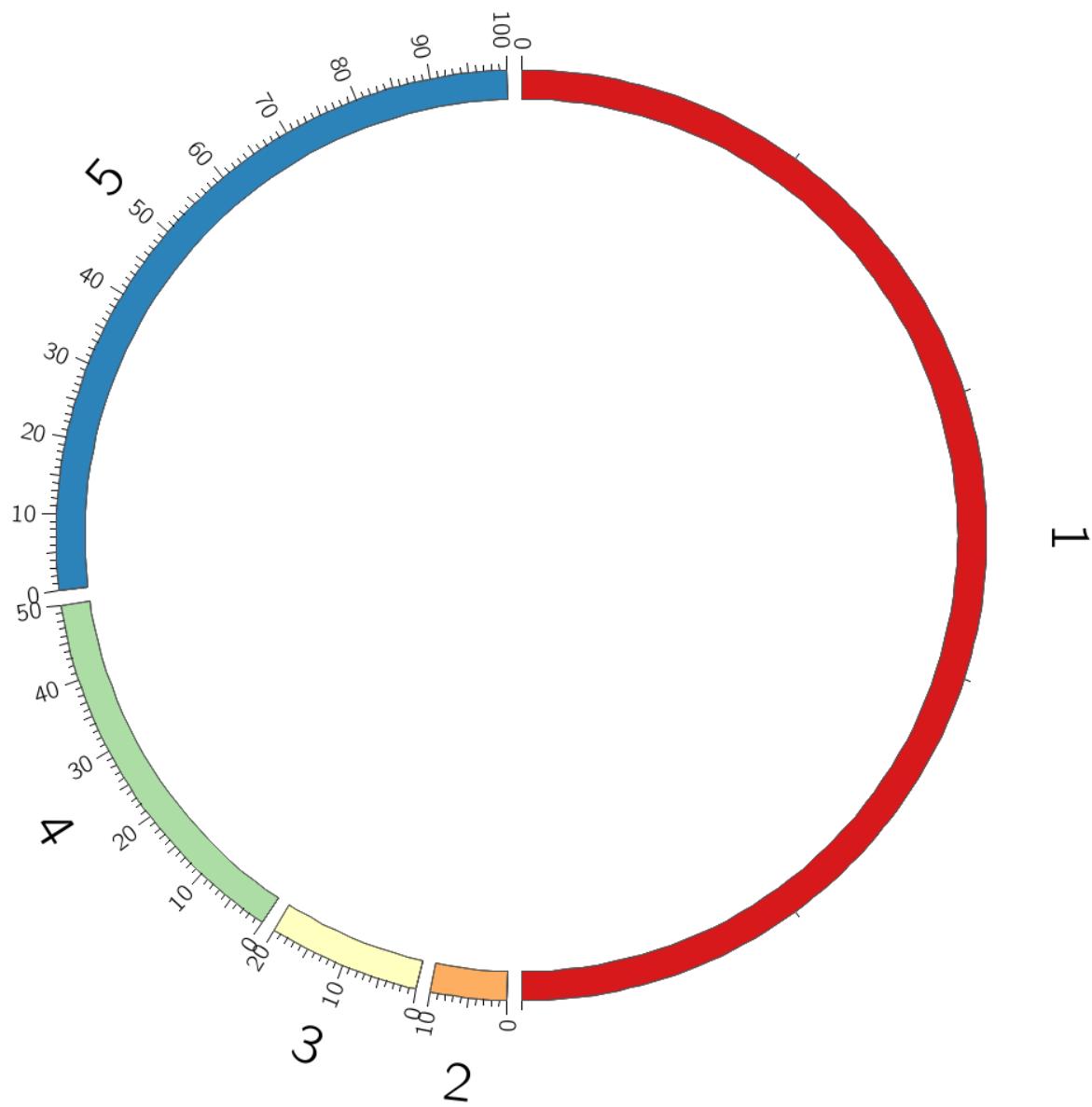


FIGURE 10

By changing the scale of chromosome 1 to $0.5r$, a relative quantity, it is made to occupy 50% of the figure.

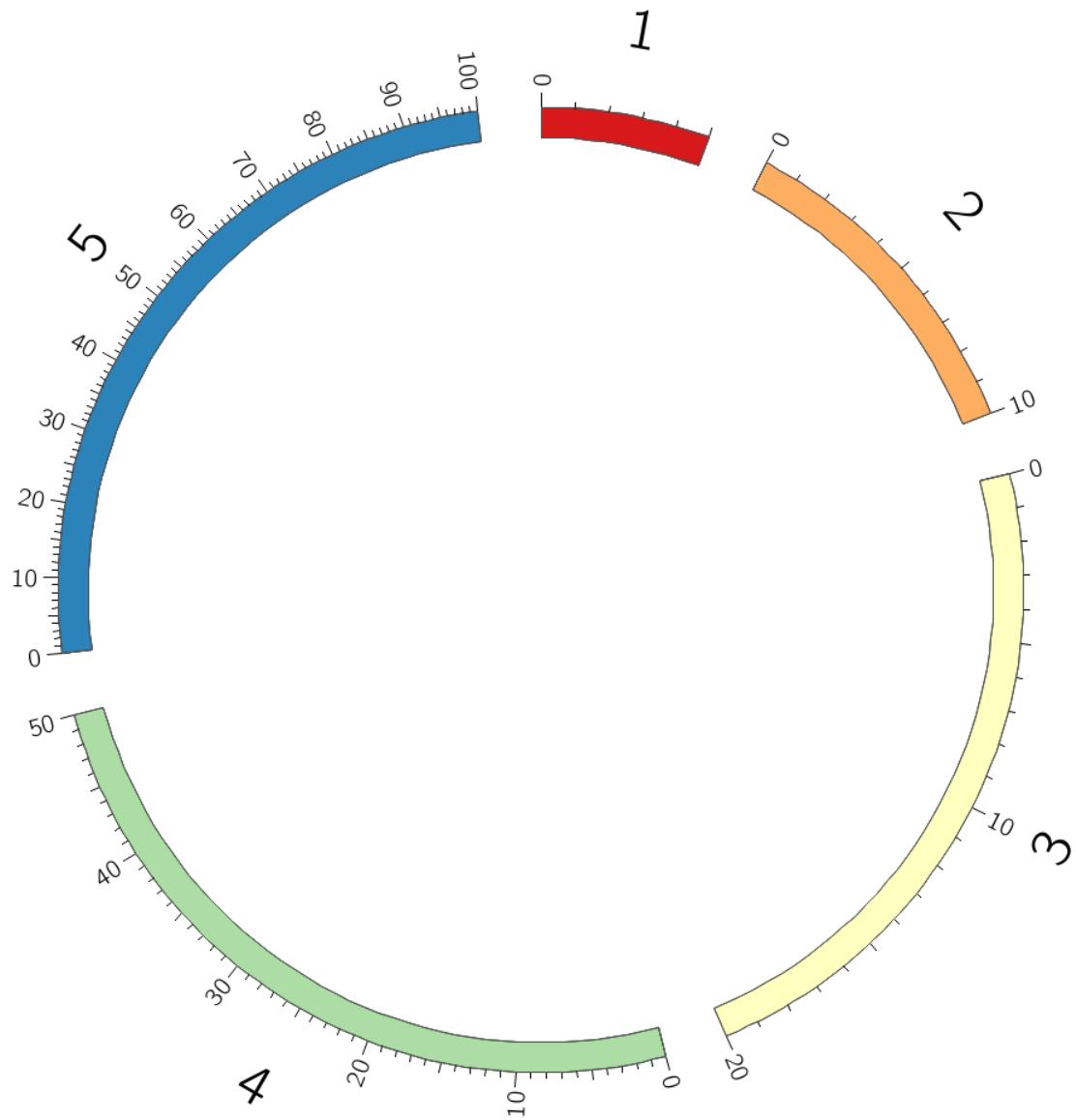


FIGURE 11

Chromosomes 4 and 5 each occupy 25% of the image. Their scale is $0.25r$.

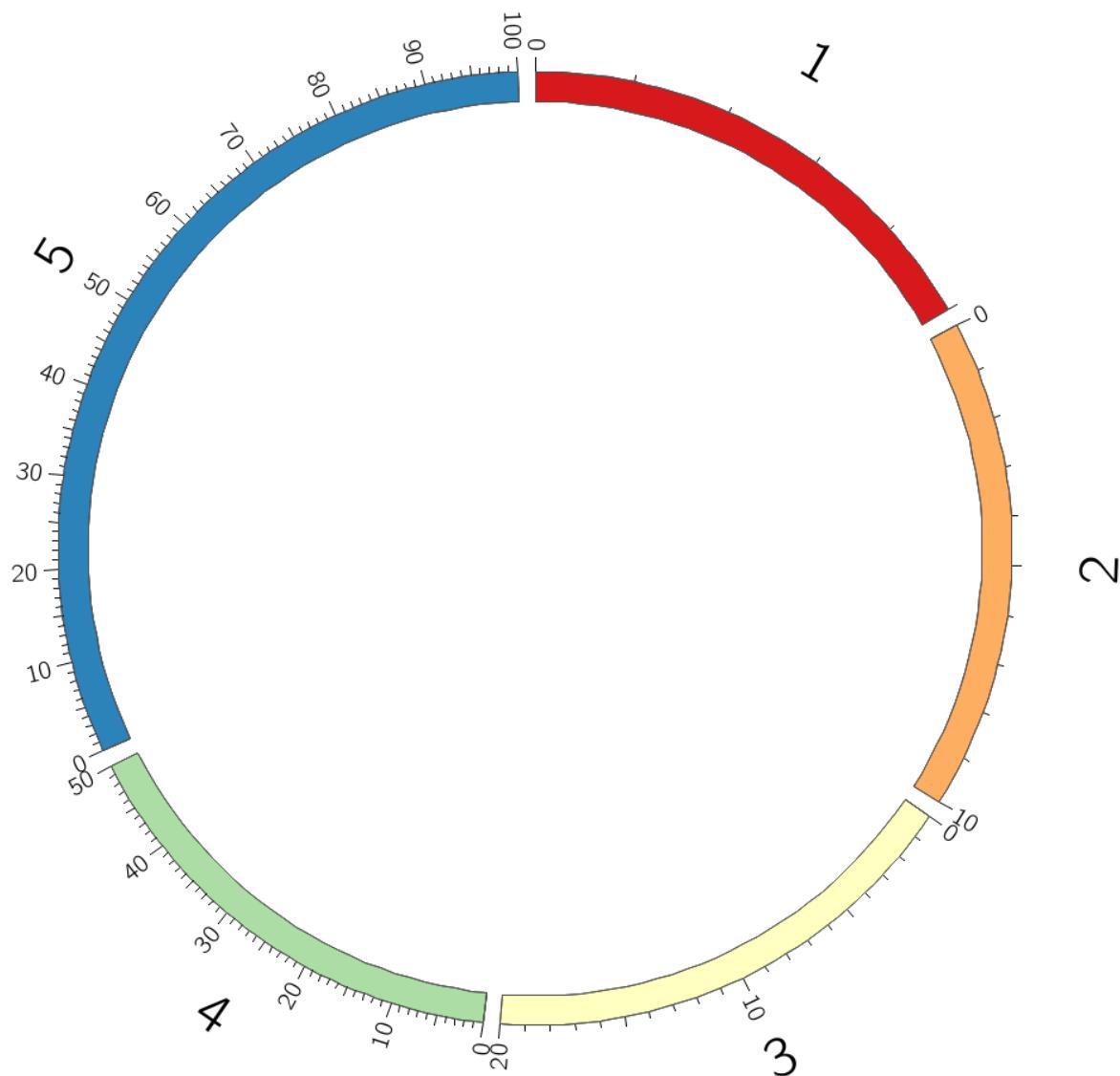


FIGURE 12

By using normalized relative scale (e.g. `0.5rn`), you can select several ideograms to be drawn to equal size within a fraction of the image. Here, chromosomes 1, 2 and 3 share 50% of the figure.

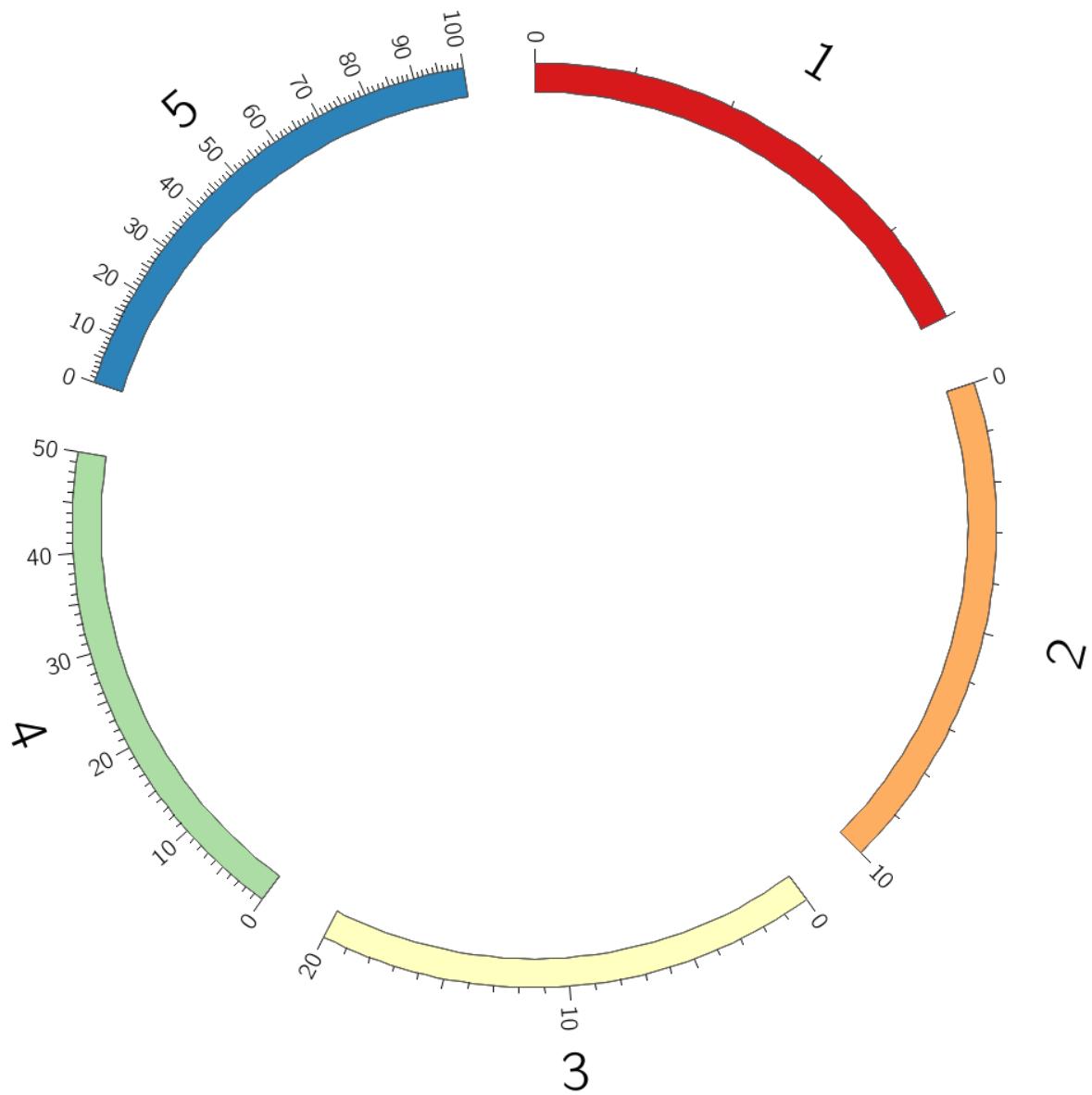


FIGURE 13

Normalized relative scale can be used to size all ideograms equally. Here, the scale is set using `/ . /=1rn`. By using a regular expression that matches every chromosome, the figure is divided evenly among the ideograms (`1rn`).

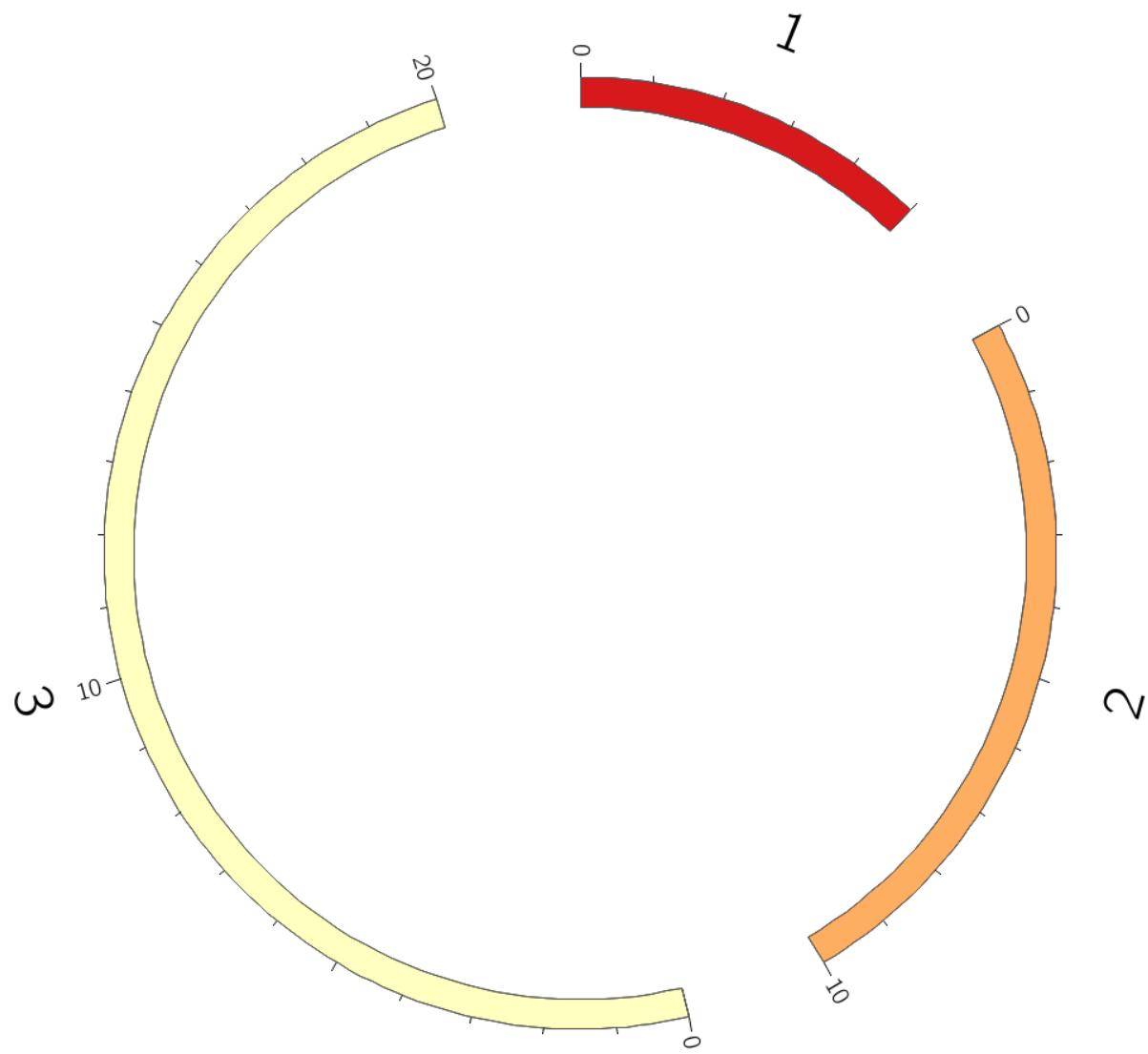


FIGURE 14

By setting `chromosomes_display_default=no` and setting the `chromosomes` parameter, you can control which chromosomes are drawn.

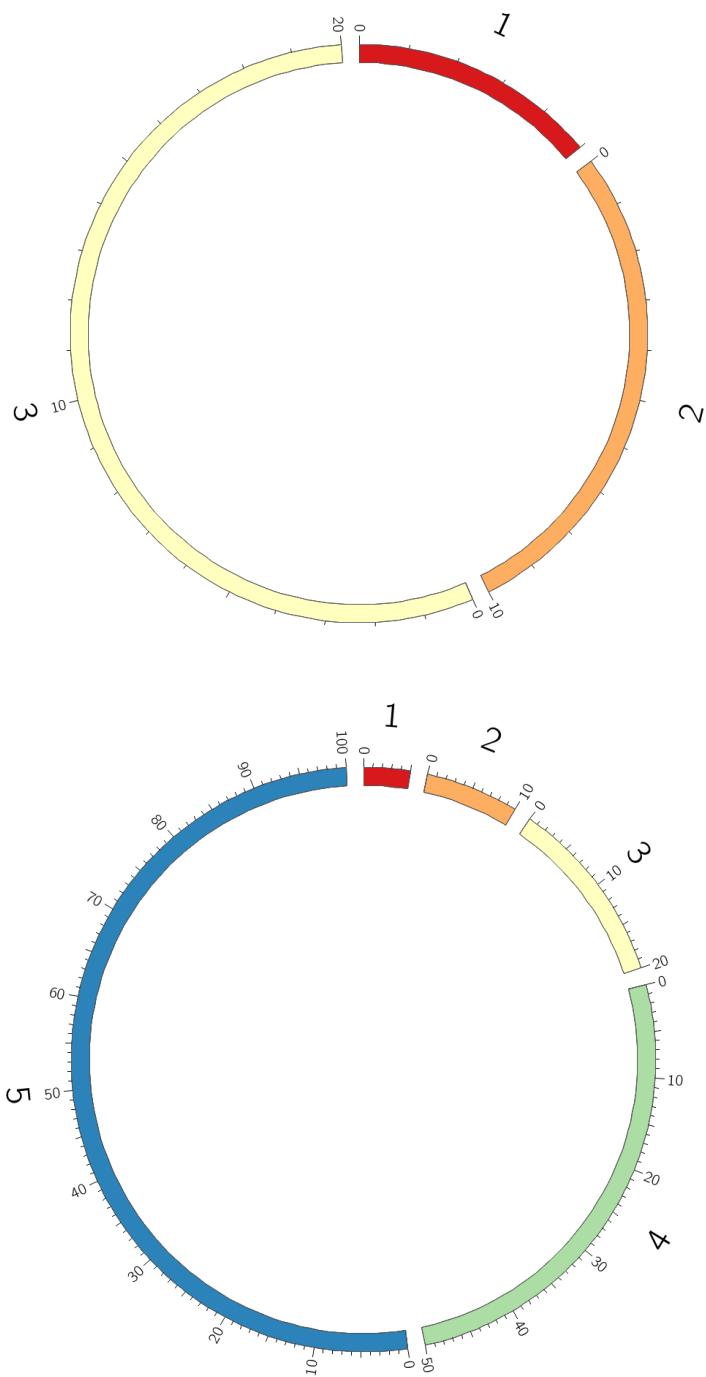


FIGURE 15

When ideogram spacing is set to relative, the fraction of the ideogram circle occupied by spaces is the same regardless of the total ideogram size.

In the top panel, spacing is set to 0.01r and the total ideogram size is 35Mb. In the bottom panel, spacing is also 0.01r but ideogram size is 185Mb.

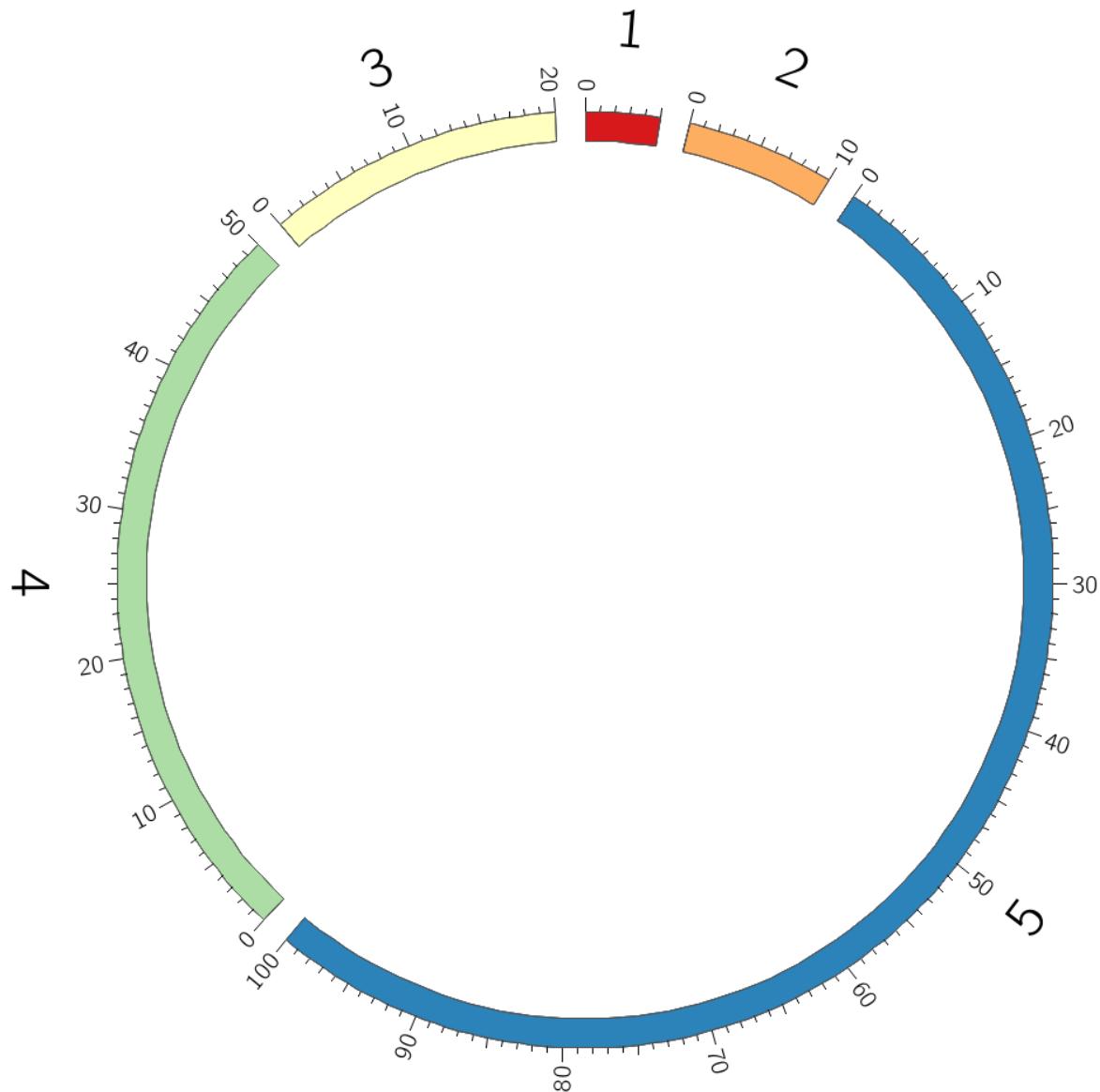


FIGURE 16

Chromosome order is adjusted using

```
chromosomes_order = chr1,chr2,chr5,chr4,chr3
```

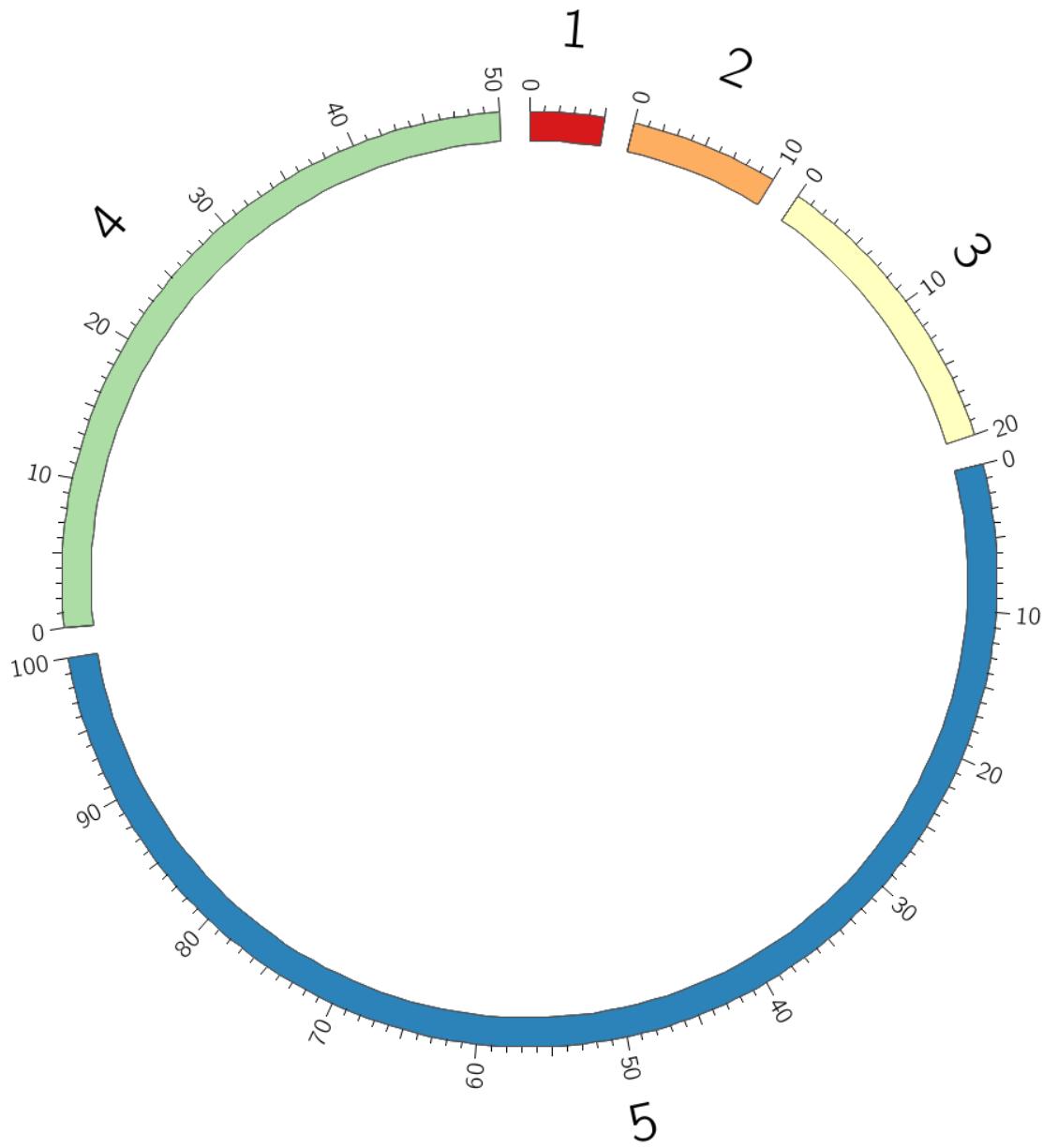


FIGURE 17

Chromosome order is adjusted using the `chromosomes_order` parameter.

To change the order of a subset of ideograms, set `chromosomes_order` to the order of the subset.
You do not need to specify the order of the full set.

```
chromosomes_order = chr3,chr5
```

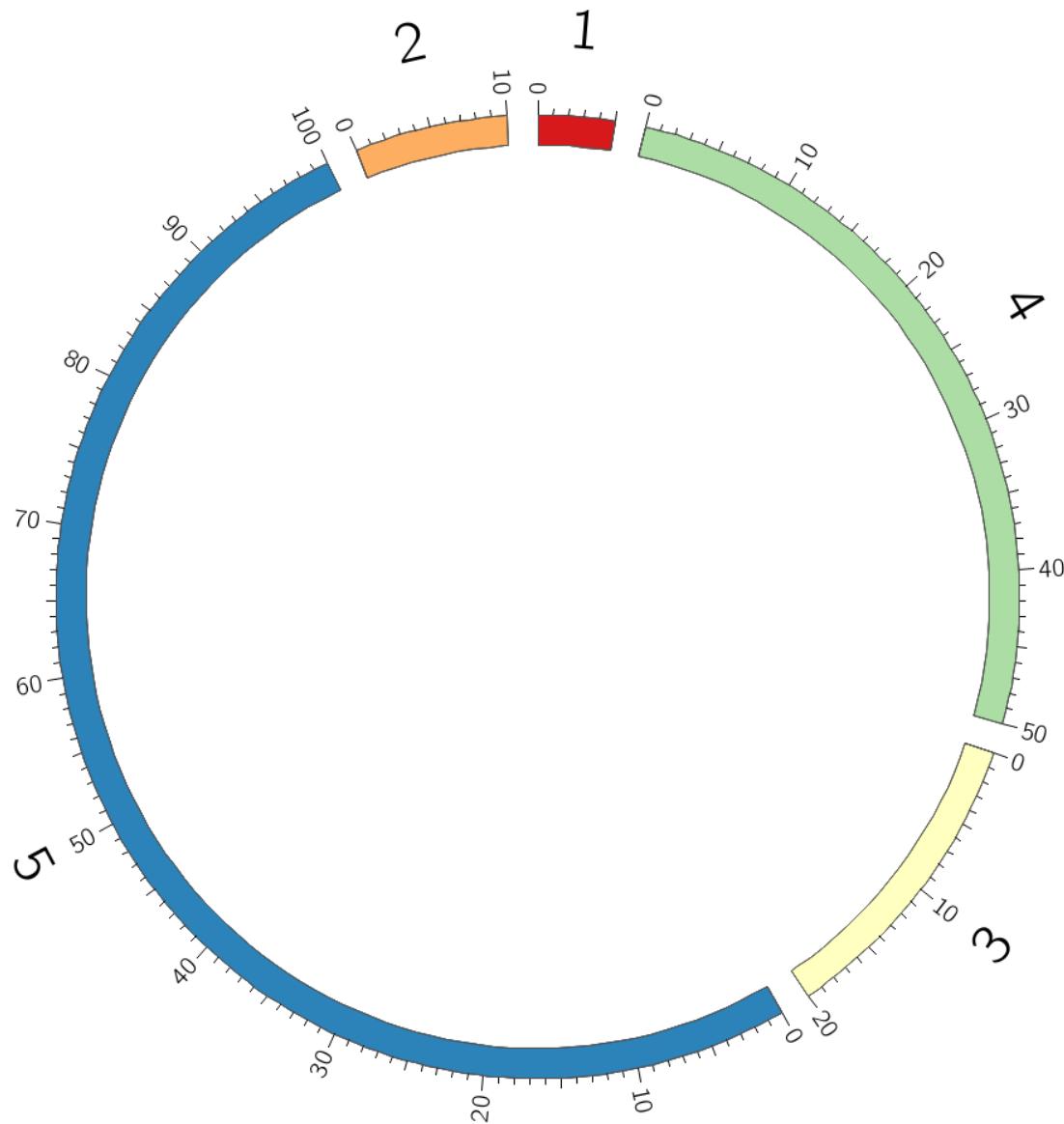


FIGURE 18

Chromosome order is adjusted using the `chromosomes_order` parameter.

To change the order of a several ideogram subsets, separate the order of subsets using “-” in `chromosomes_order`.

```
chromosomes_order = chr1,chr4,-,-,chr2
```

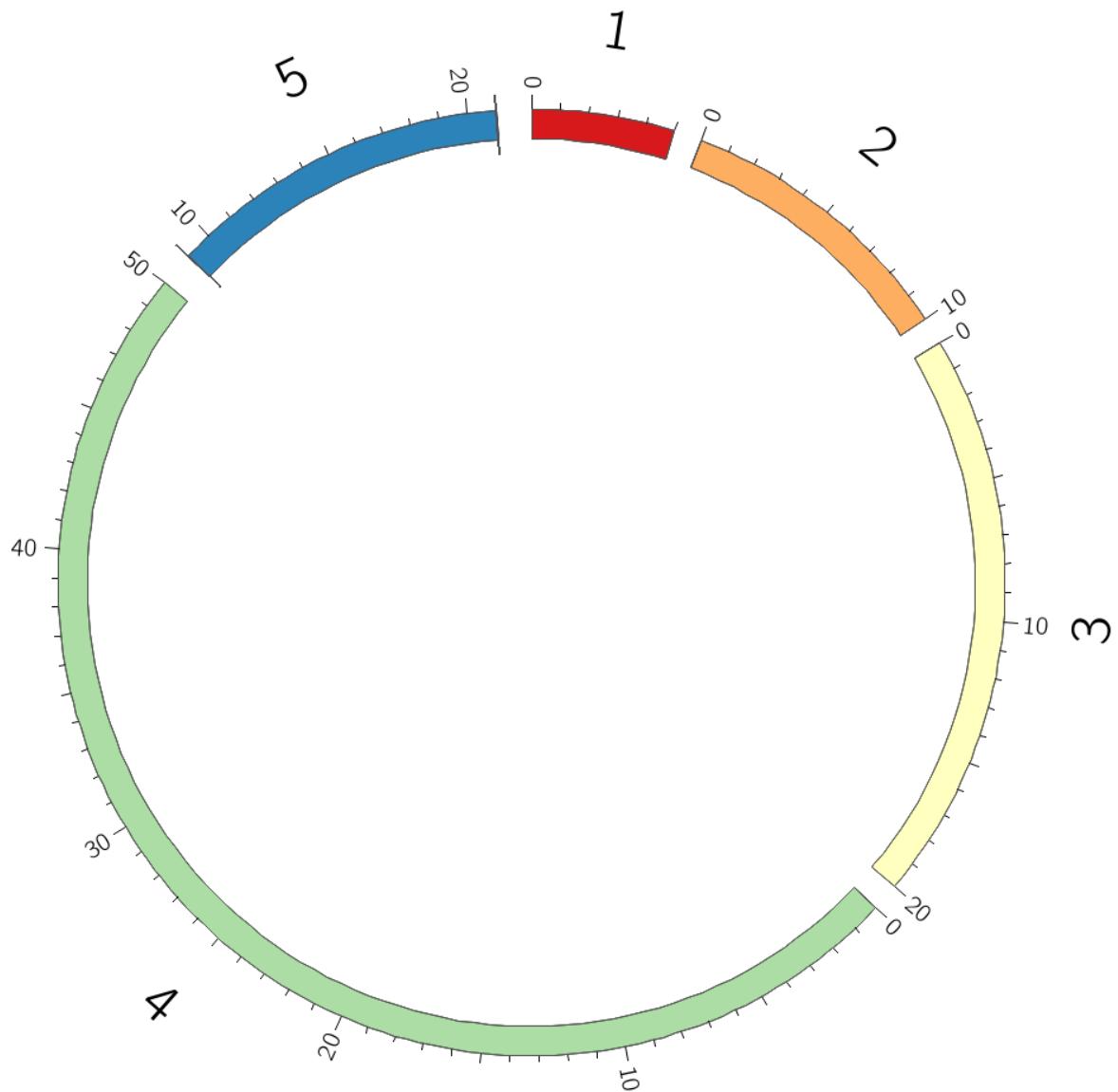


FIGURE 19

Ideograms can be cropped by using the `chromosomes` parameter.

`chromosomes = chr5:9-21`

which results in the ideogram of chromosome 5 cropped to the 9-21Mb region.

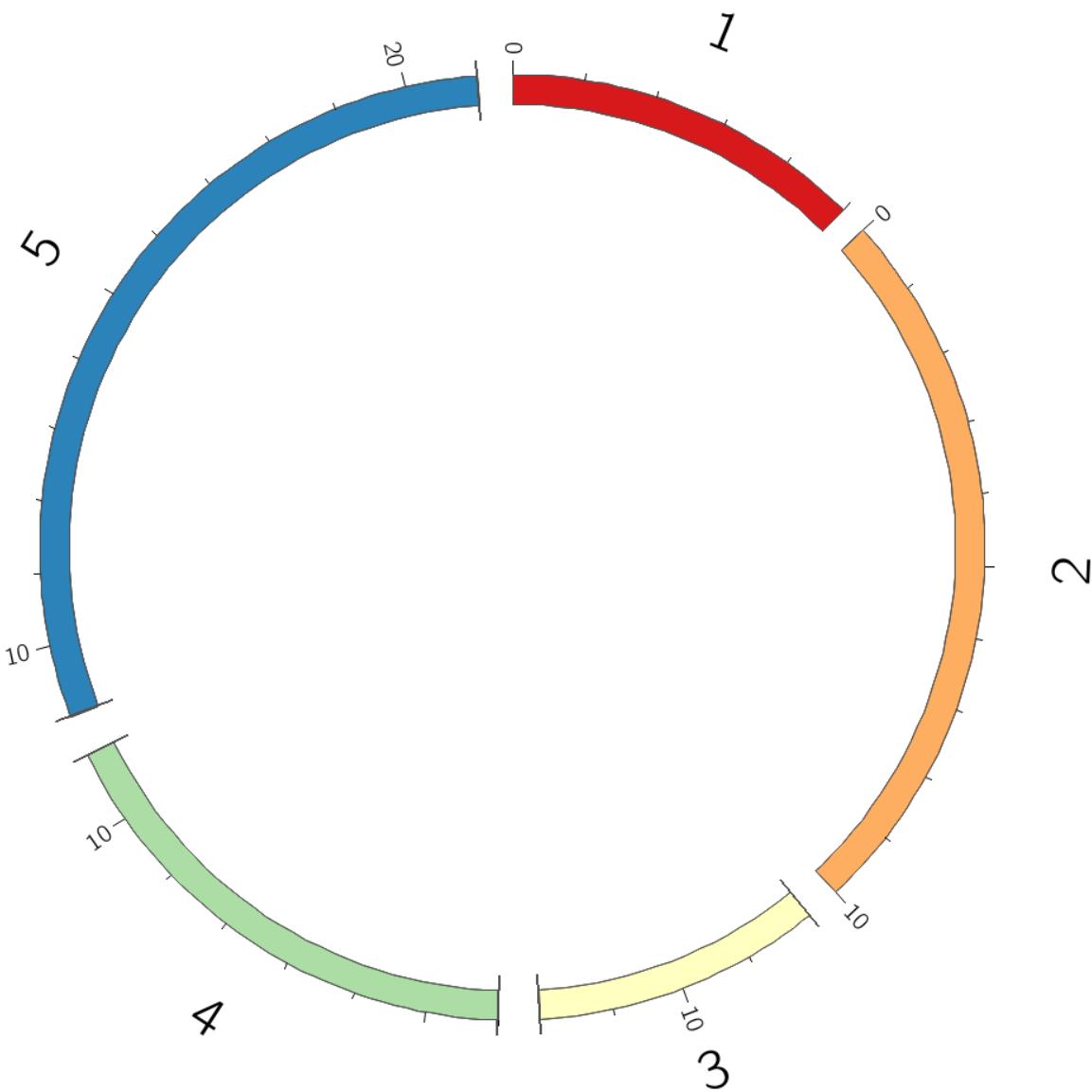


FIGURE 20

Multiple ideograms can be cropped by using the `chromosomes` parameter, which results in the ideograms of chromosomes 3, 4 and 5 being cropped.

```
chromosomes = chr3:8-12;chr4:4-11;chr5:9-21
```

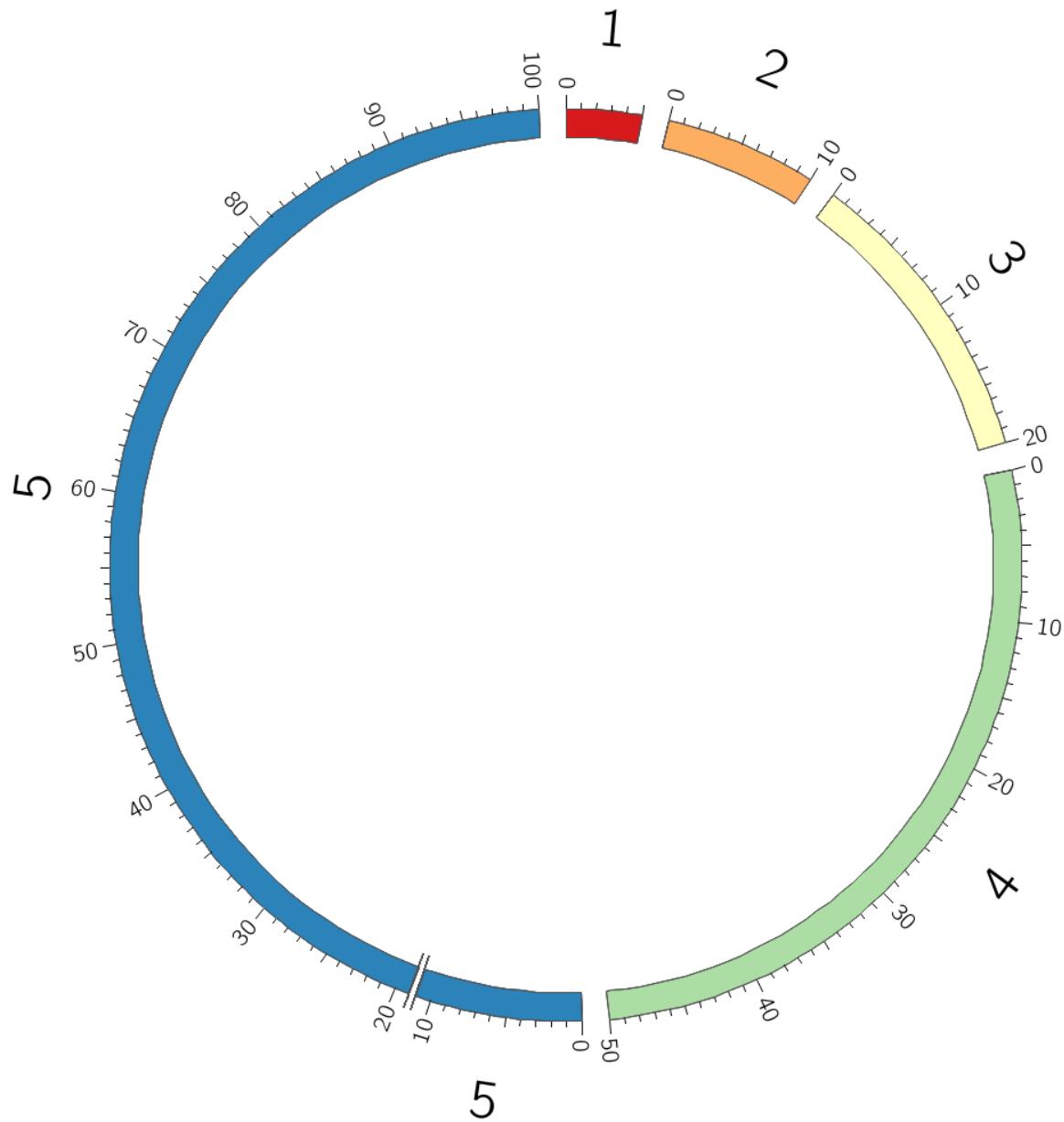


FIGURE 21

Regions of chromosomes can be removed from the display by using `chromosomes_breaks`.

```
chromosomes_breaks = -chr5:11-19
```

which results in a figure in which the region 11-19Mb of chromosome 5 is missing. In place of this region, an axis break is shown.

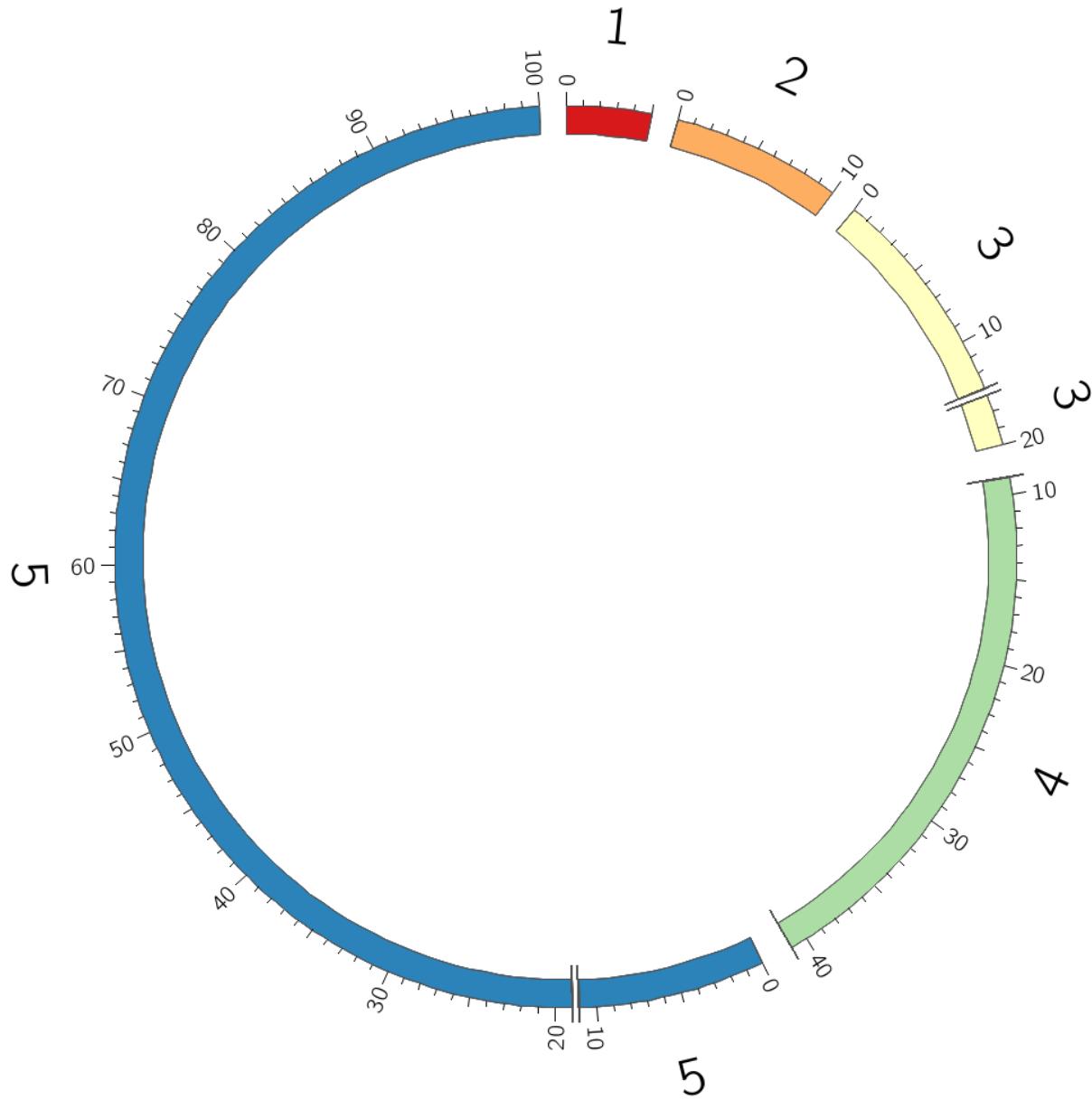


FIGURE 22

Multiple breaks can be defined by including more than one region to remove from the figure in `chromosomes_breaks`.

```
chromosomes_breaks = -chr3:13-17;-chr4:(-9;-chr4:41-);-chr5:11-19
```

which results in breaks in chromosomes 3, 4 (two breaks) and 5.

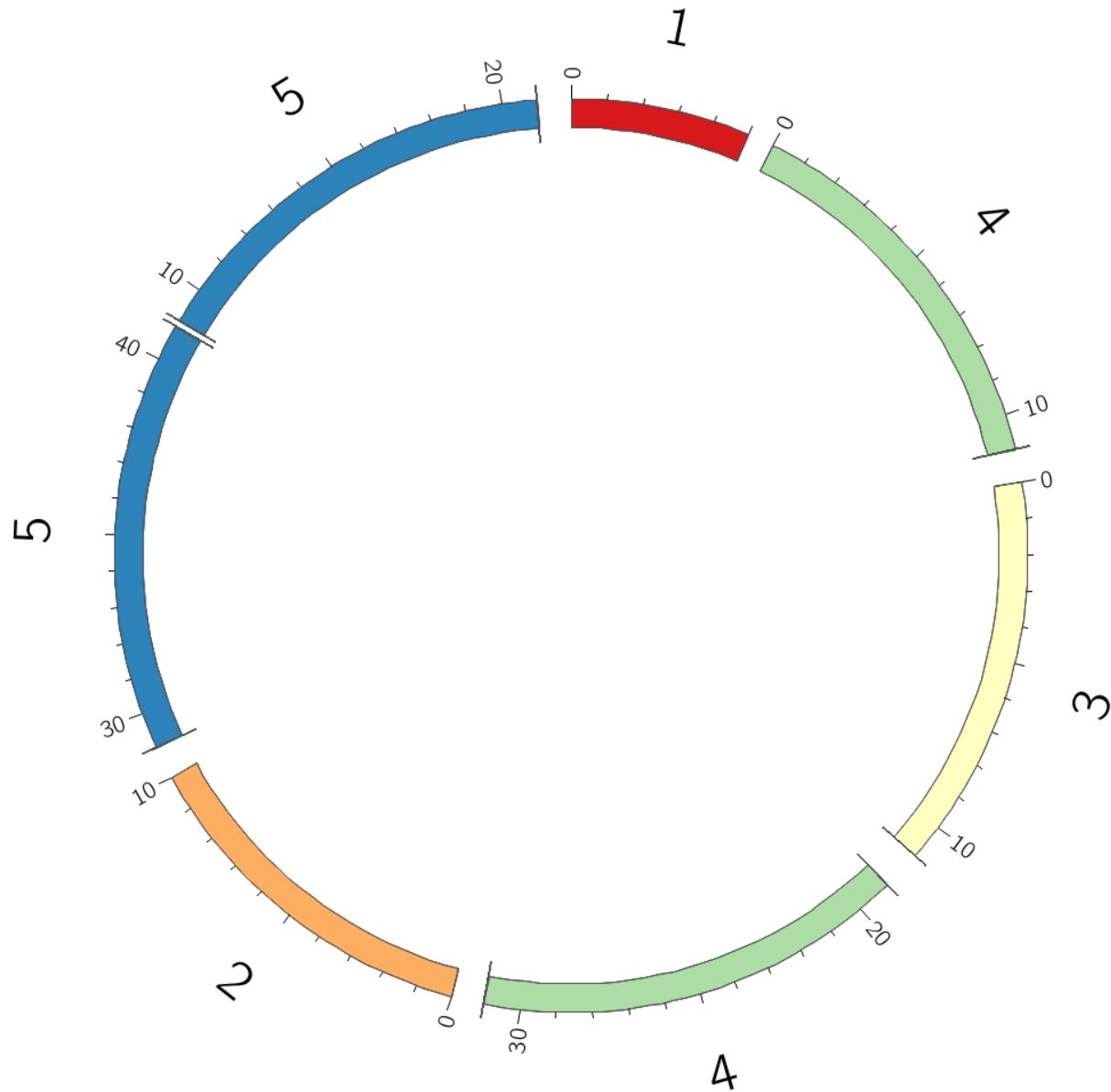


FIGURE 23

A chromosome can be split into multiple, independently ordered ideograms by using tags in the `chromosomes` parameter to associate a unique name to each ideogram.

```
chromosomes = chr3:0-11;chr4[4a]:0-11;chr4[4b]:19-31;chr5[5a]:9-21;chr5[5b]:29-41
```

Chromosome 4 is split into two pieces (a and b) which correspond to 0-11Mb and 19-31Mb of the chromosome.

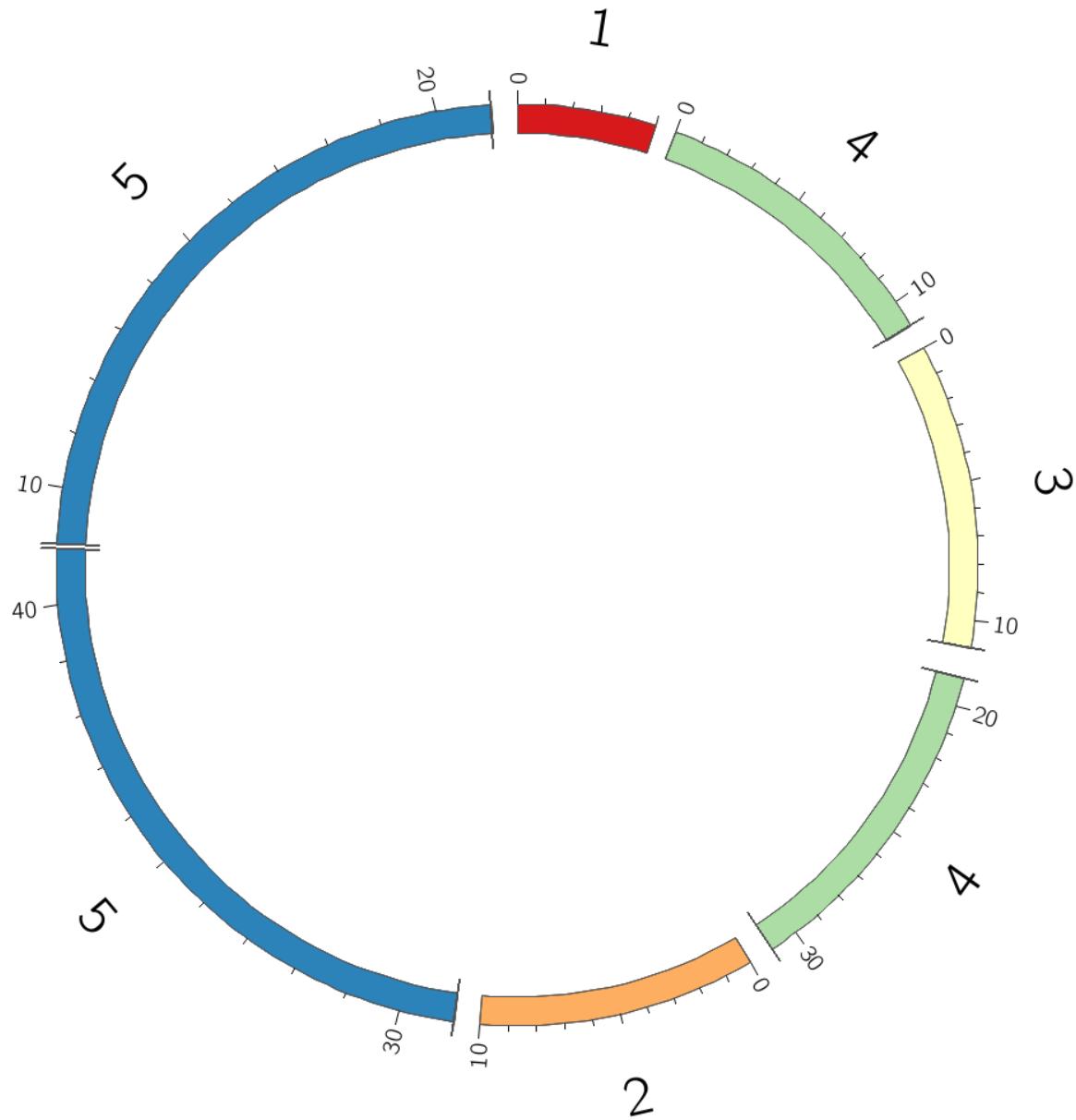


FIGURE 24

Ideograms formed by breaking up chromosomes can be scaled.

```
chromosomes = chr3:0-11;chr4[4a]:0-11;chr4[4b]:19-31;chr5[5a]:9-21;chr5[5b]:29-41
```

```
chromosomes_scale = 5a=2,5b=2
```

A

COLOR	RGB
gpos100	0,0,0
gpos	0,0,0
gpos75	130,130,130
gpos66	160,160,160
gpos50	200,200,200
gpos33	210,210,210
gpos25	200,200,200
gvar	220,220,220
gneg	255,255,255
acen	217,47,39
stalk	100,127,164

B

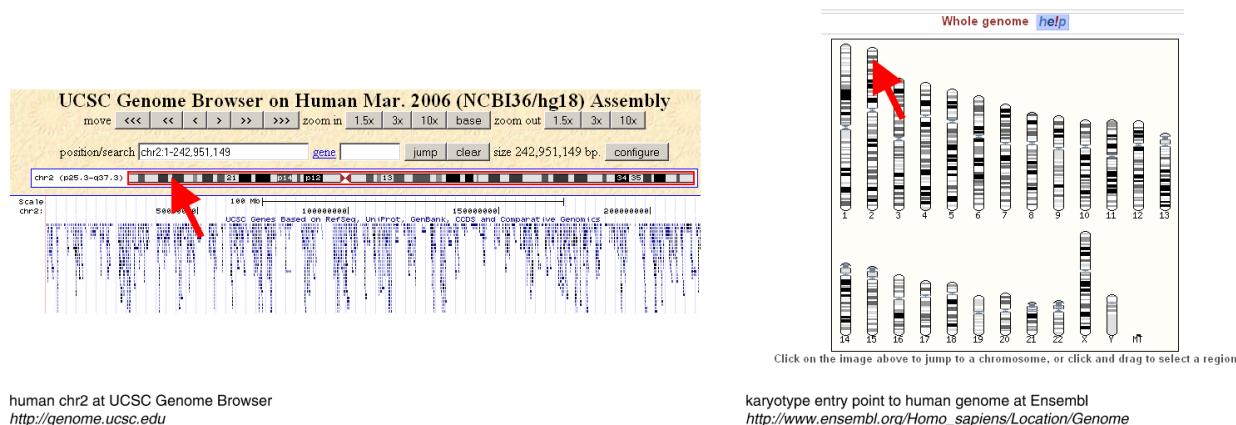


FIGURE 25

(A) Band colors are defined in `etc/color.ucsc.conf` in the Circos distribution directory to correspond to the color scheme of the UCSC genome browser and the names given to the bands in the UCSC Karyotype table. For example, acen is used for centromeres, typically shown in red. (B) views showing cytogenetic bands in UCSC and Ensembl browsers (same band position is indicated by a red arrow).

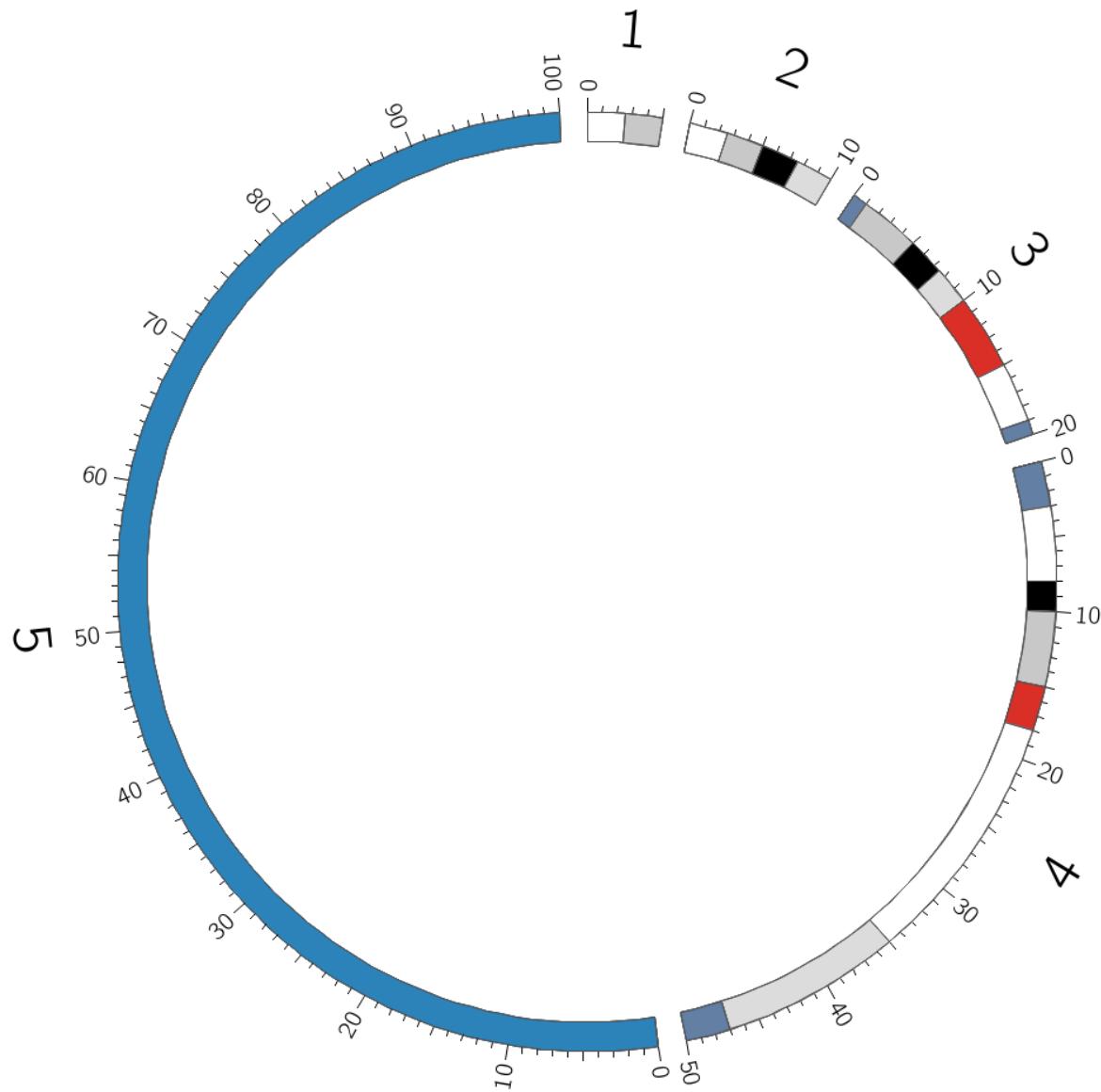


FIGURE 26

Cytogenetic bands appear as stripes within the ideogram segment.

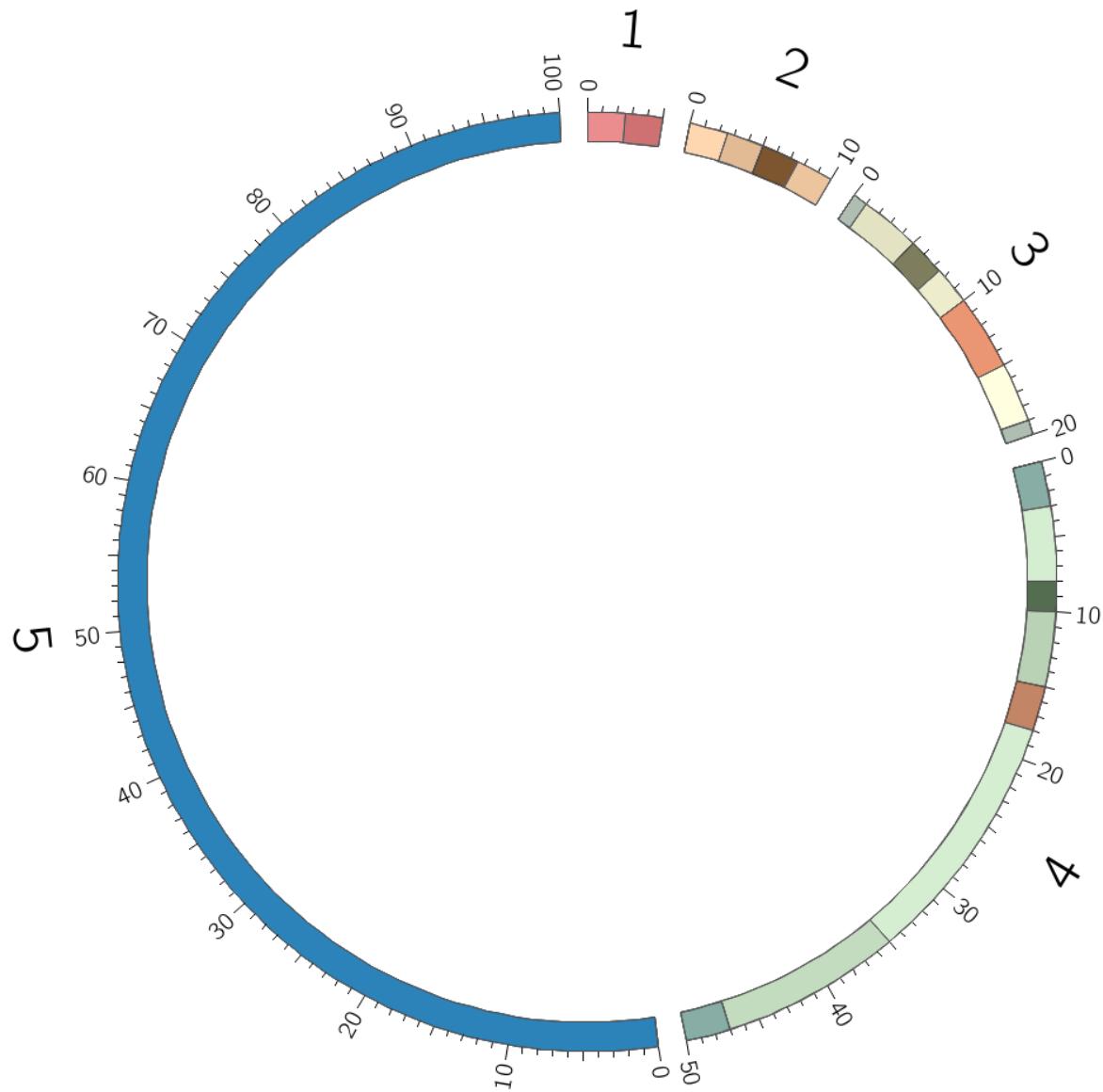


FIGURE 27

By changing `band_transparency` in the `<ideogram>` block, the band pattern can be made semi-transparent to allow the color of the ideogram to show through.

Using this parameter you can combine the ideogram color and the band pattern.

In this example, `band_transparency = 3`.

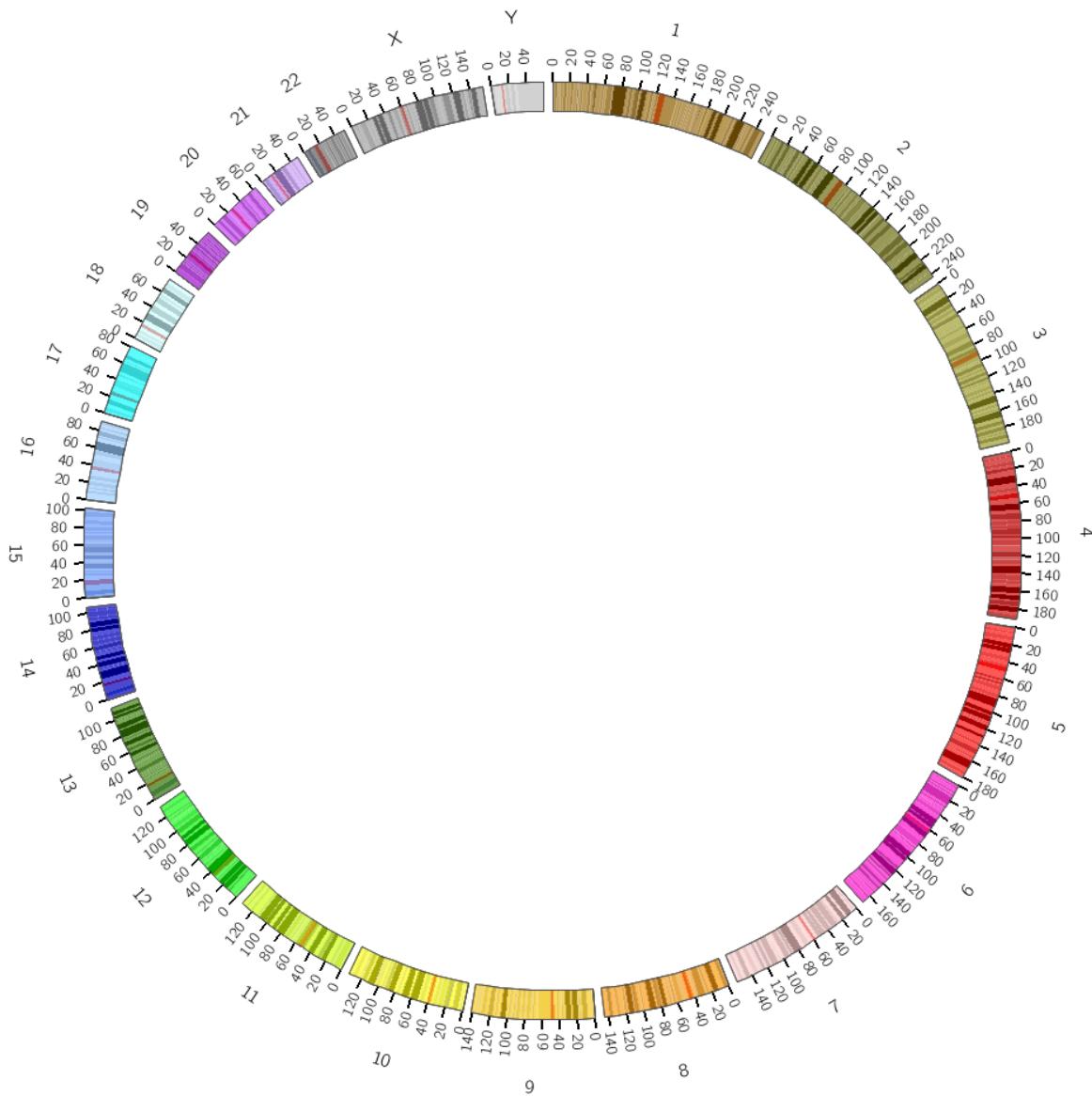


FIGURE 28

All chromosomes in the human genome. The figure uses the human karyotype file which defines 24 chromosomes (1..22, X, Y) and the cytogenetic bands.

There is a conventional color palette for the human genome. These colors are defined in the `color.conf` file, named after the chromosome name, but with a “`chr`“ prefix (e.g. `hs1` has color `chr1`).

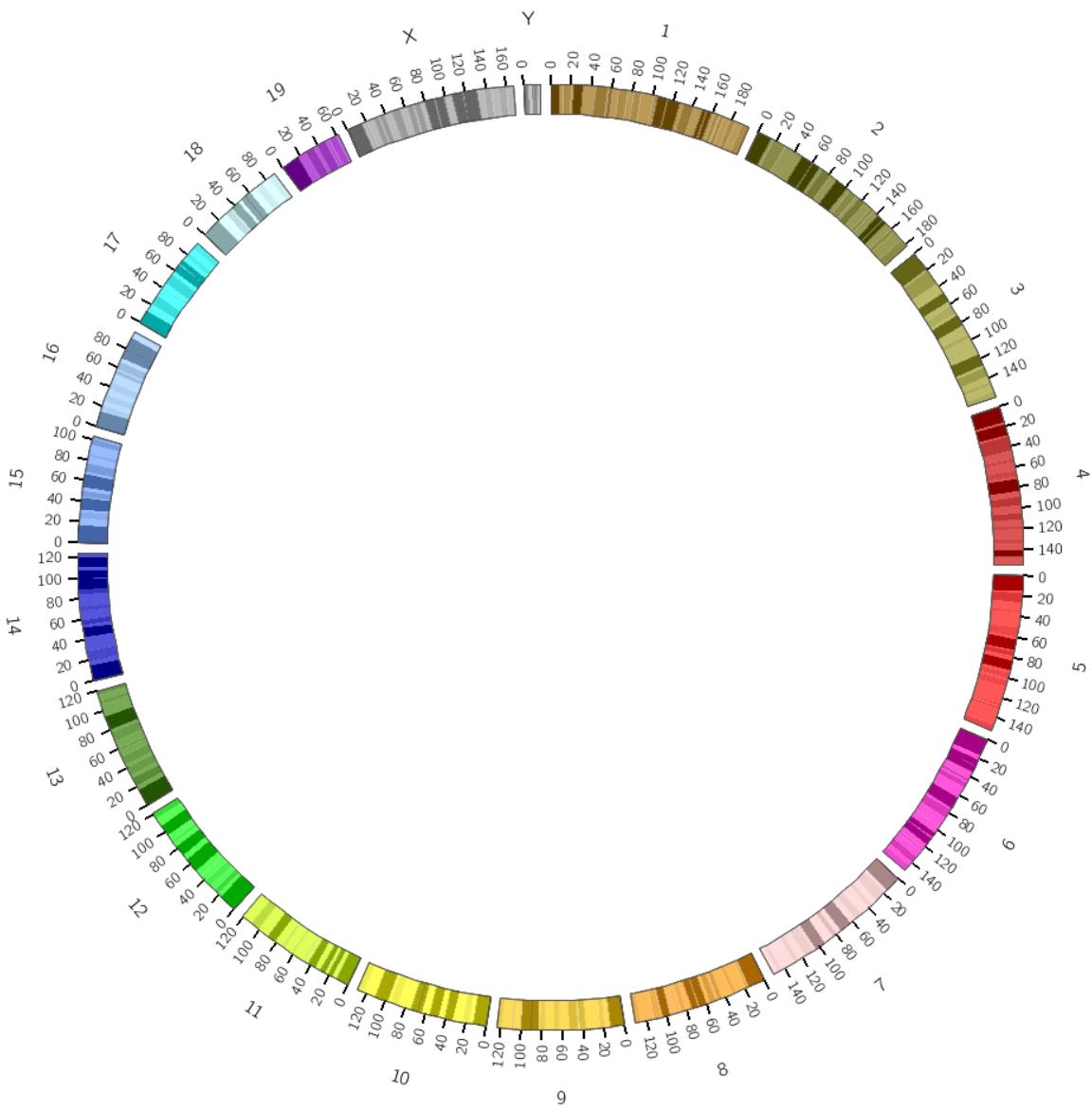


FIGURE 29

All chromosomes in the mouse genome. The figure uses the human karyotype file which defines 21 chromosomes (1..19, X, Y) and the cytogenetic bands.

UCSC GENOME BROWSER
HUMAN CHROMOSOME
COLOR PALETTE



FIGURE 30

Conventional color palette for the human genome used by the UCSC genome browser. These colors are defined in the `etc/color.ucsc.conf` file in the Circos distribution, named after the chromosome name, but with a 'chr' prefix (e.g. `hs1` has color `chr1`).

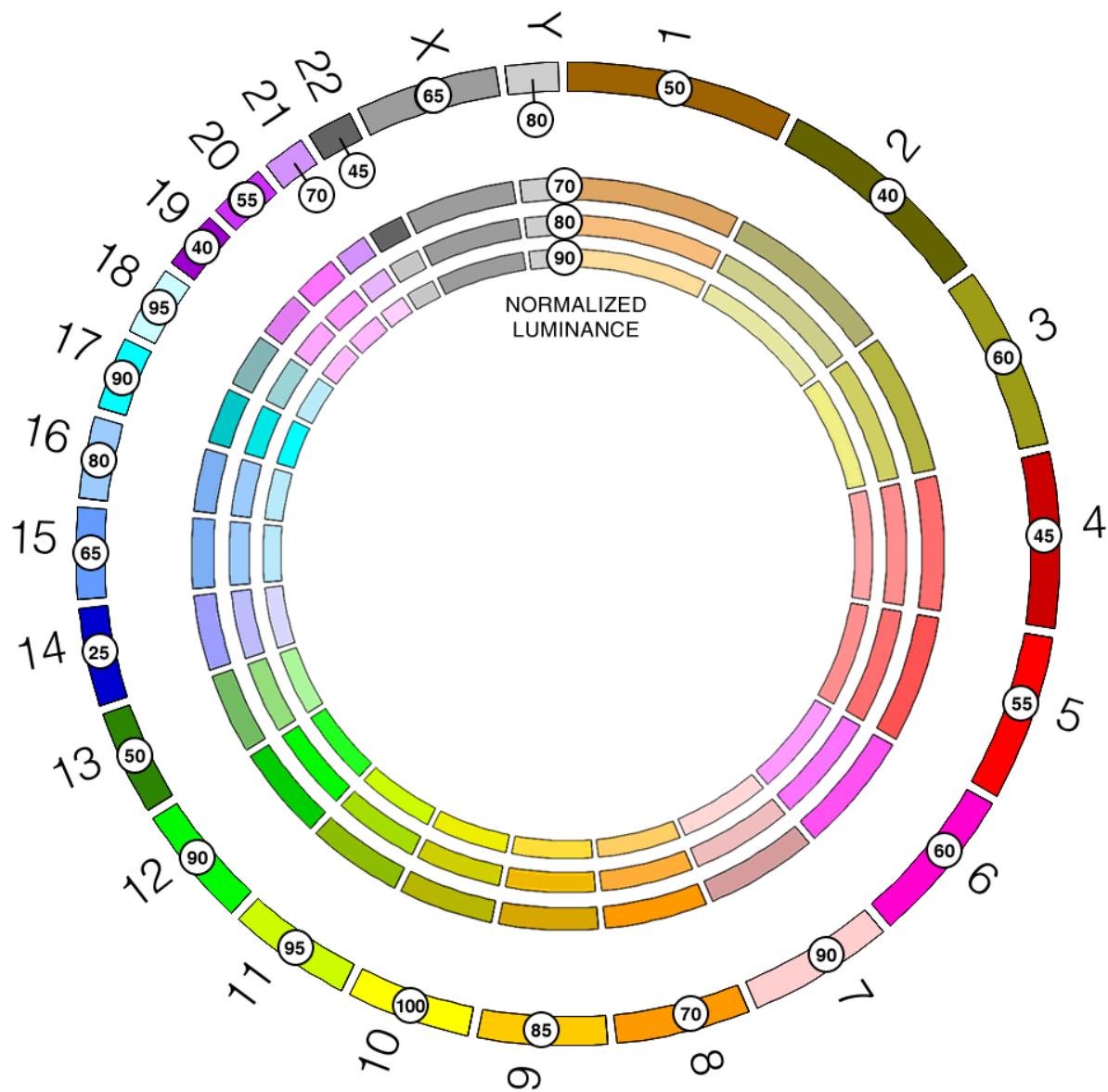


FIGURE 31

The conventional UCSC color scheme (outer ring) with luminance values (perceived brightness) of each color. Luminance-normalized schemes for luminance 70, 80 and 90 are shown inside the figure.

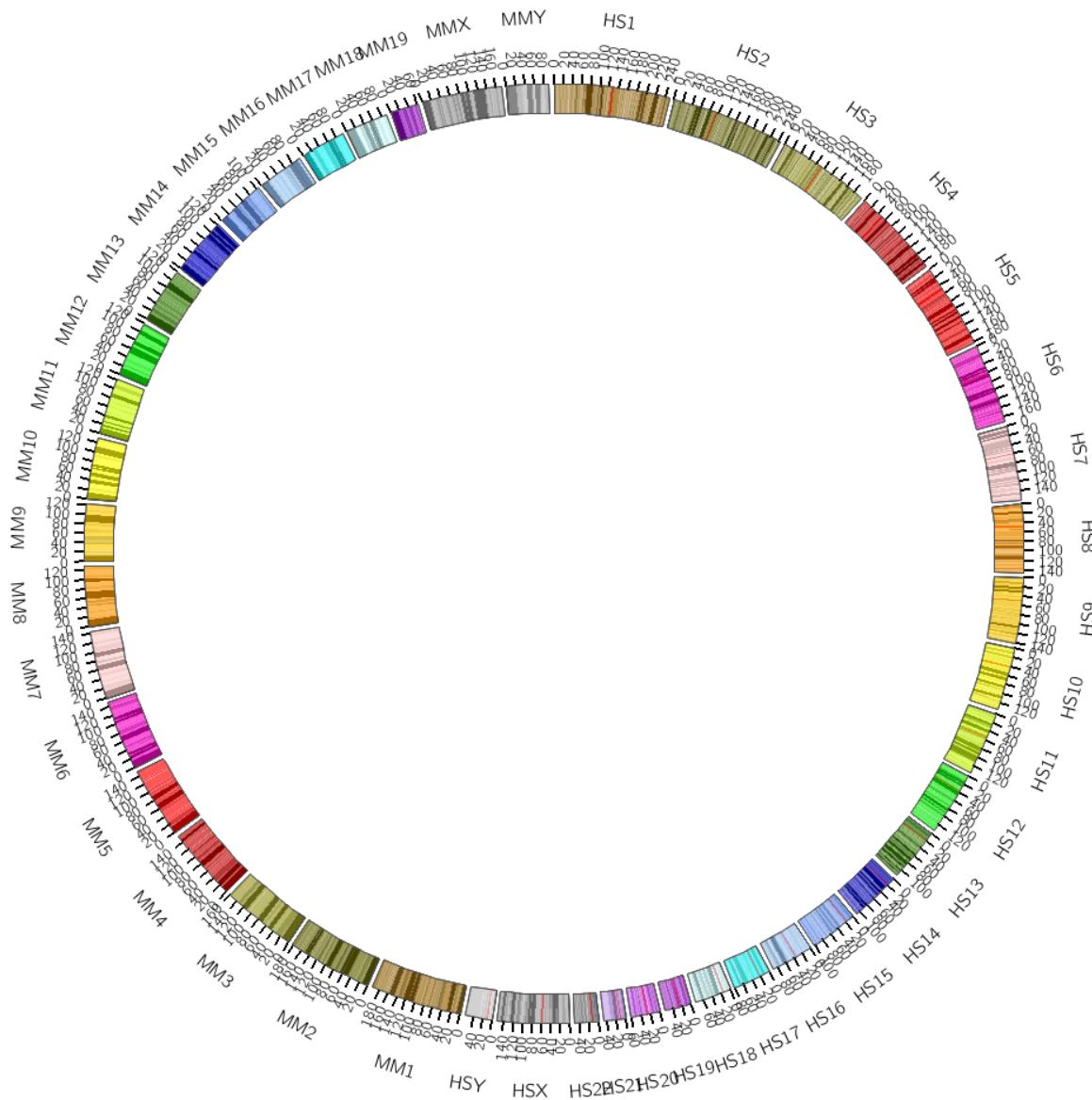


FIGURE 32

Ideograms of all human (labeled `hsN`) and mouse (labeled `mmN`) chromosomes. Human ideograms are shaded by the UCSC genome browser color convention. The mouse chromosomes do not have a conventional color assignment and are colored white using `chromosomes_color=/mm/=white`.

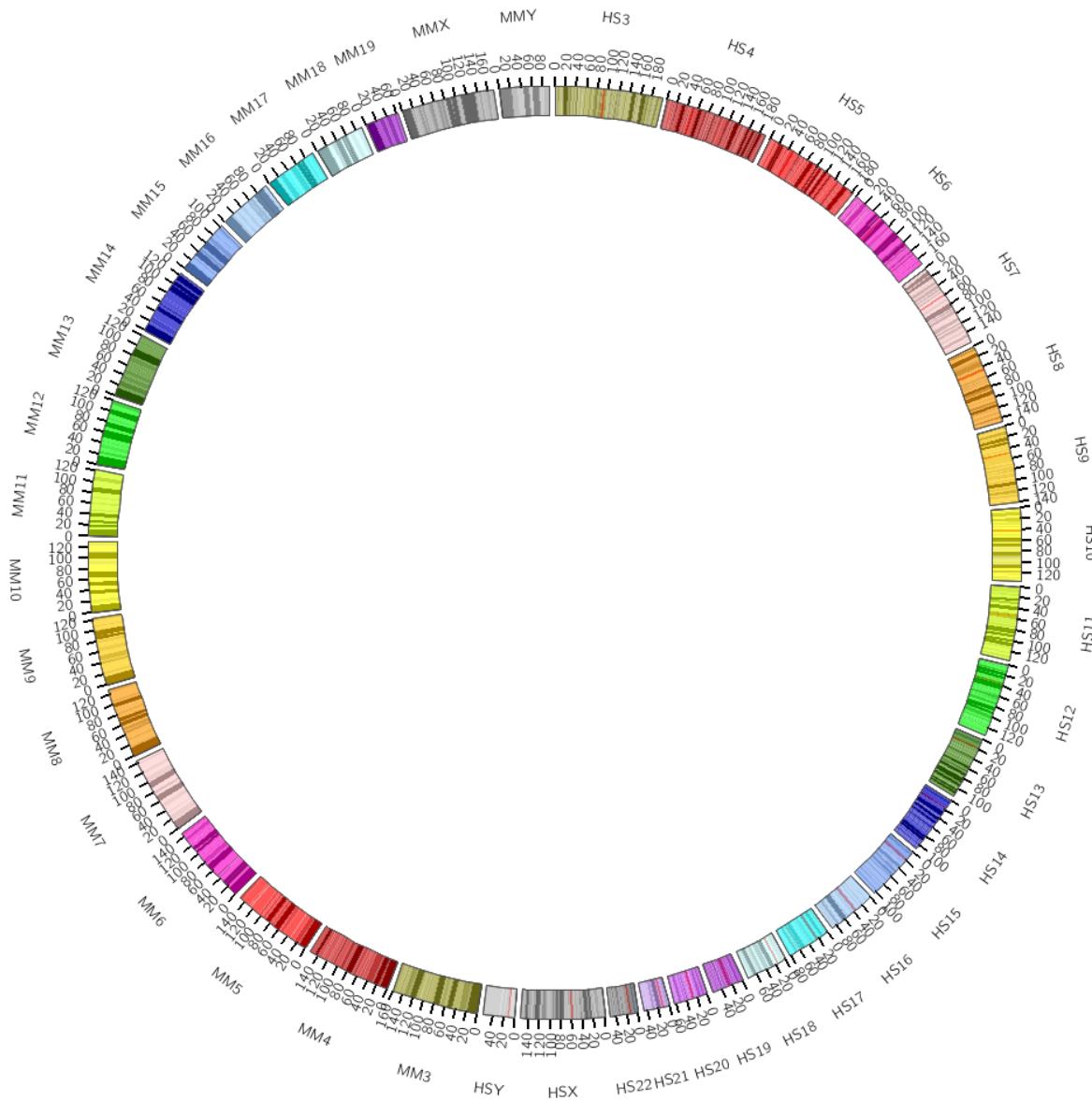


FIGURE 33

Chromosomes can be removed from the figure by using “-“ prefix in the chromosomes parameter. Here, hs1 hs2 mm1 mm2 were removed using

```
chromosomes_display_default = yes
chromosomes = -hs1;-hs2;-mm1;-mm2
```

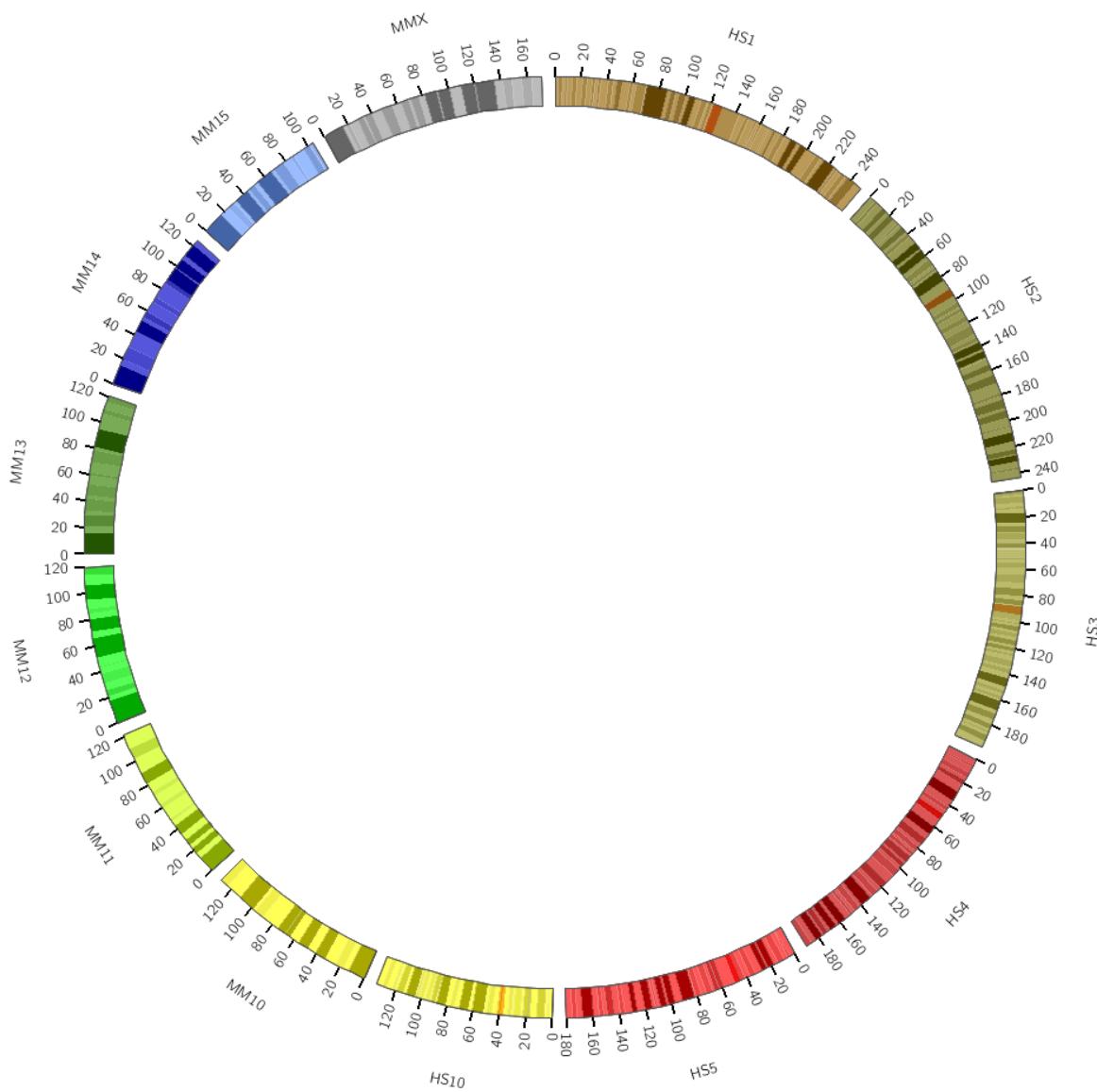


FIGURE 34

When `chromosomes_display_default=no`, you can specifically define which ideograms to display. Here, using a combination of regular expressions and ideogram names, hs1...hs5, hs10, mm10...mmx and mmx are shown.

```
chromosomes_display_default = no
chromosomes = /hs[1-5]$;/hs10;/mm1[0-5]/;mmx
```

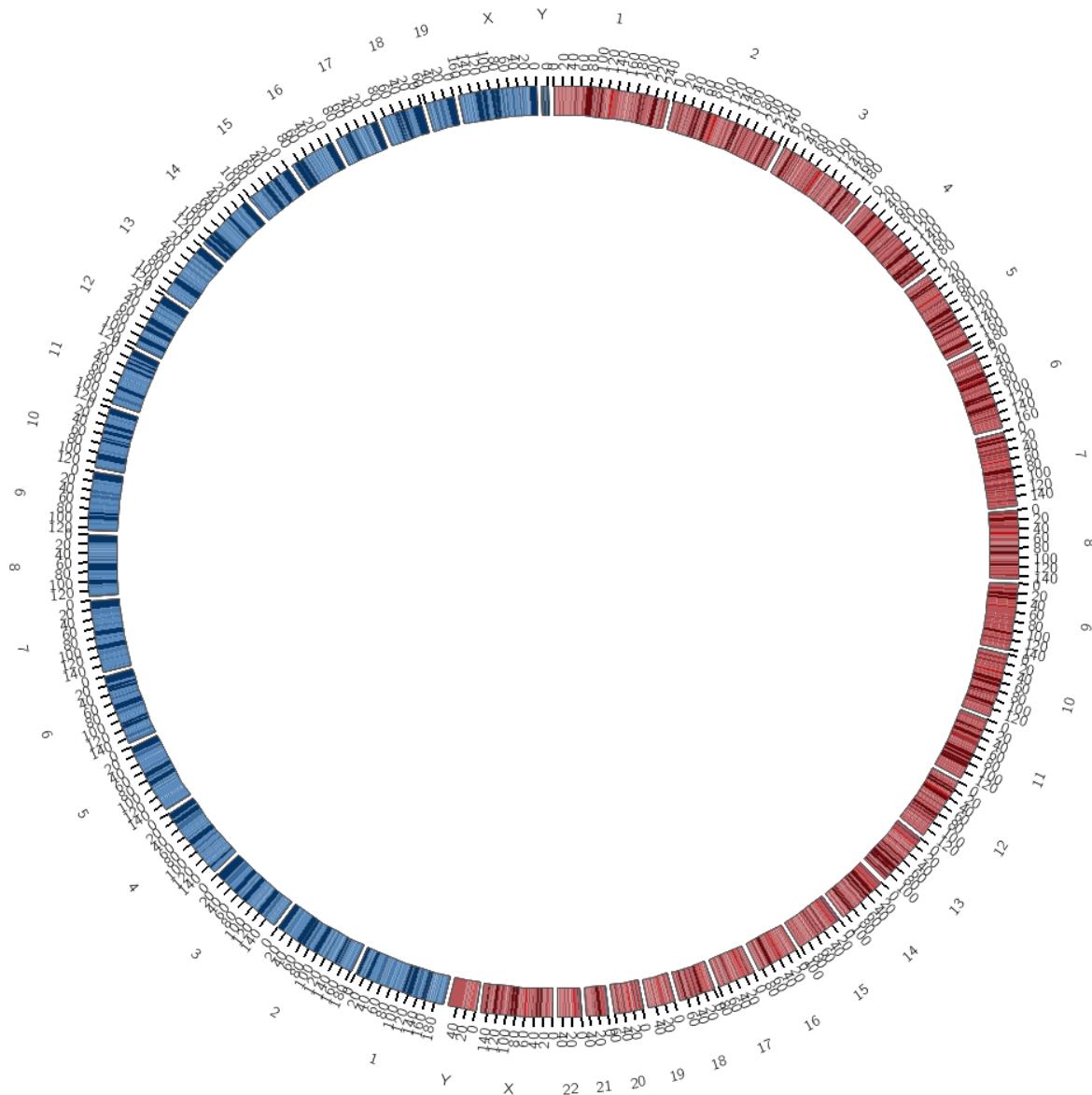


FIGURE 35

The color of ideogram can be changed using `chromosomes_color`. Here human chromosomes are made red and those of the mouse, blue.

```
chromosomes_color = /hs/=reds-5-seq-5,/mm=blues-5-seq-5
```

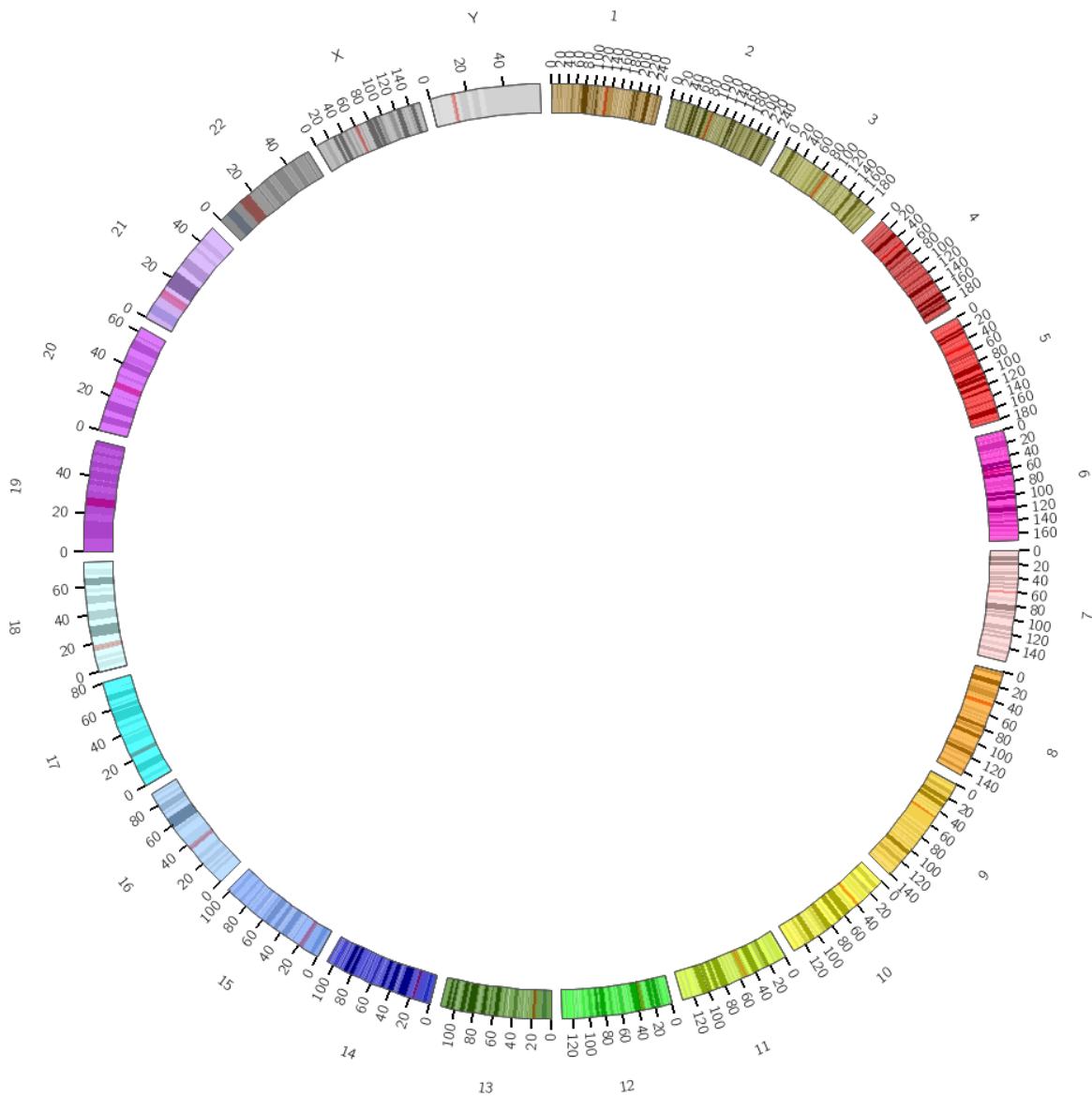


FIGURE 36

By setting the same relative scale to all ideograms, their length is normalized. Here, each ideogram occupies 1/24th of the image using normalized relative scale.

```
chromosomes_scale = ./=1rn
```

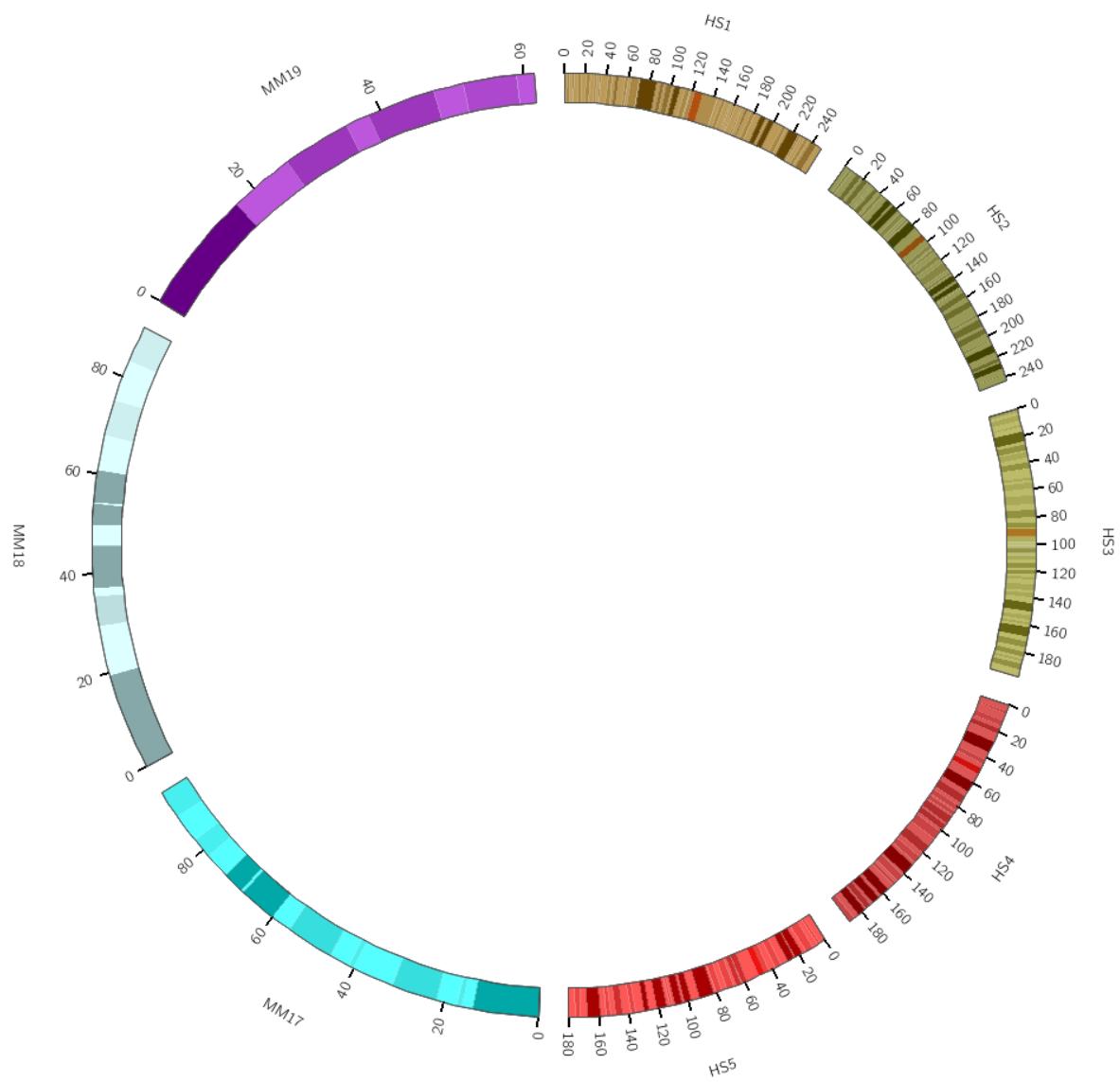


FIGURE 37

Complex symmetric layouts can be created by setting different relative scales to ideogram groups. Here, 5 human and 3 mouse ideograms are shown. By setting the scale for human and mouse ideograms to $0.5\pi n$, their groups occupy 50% of the image.

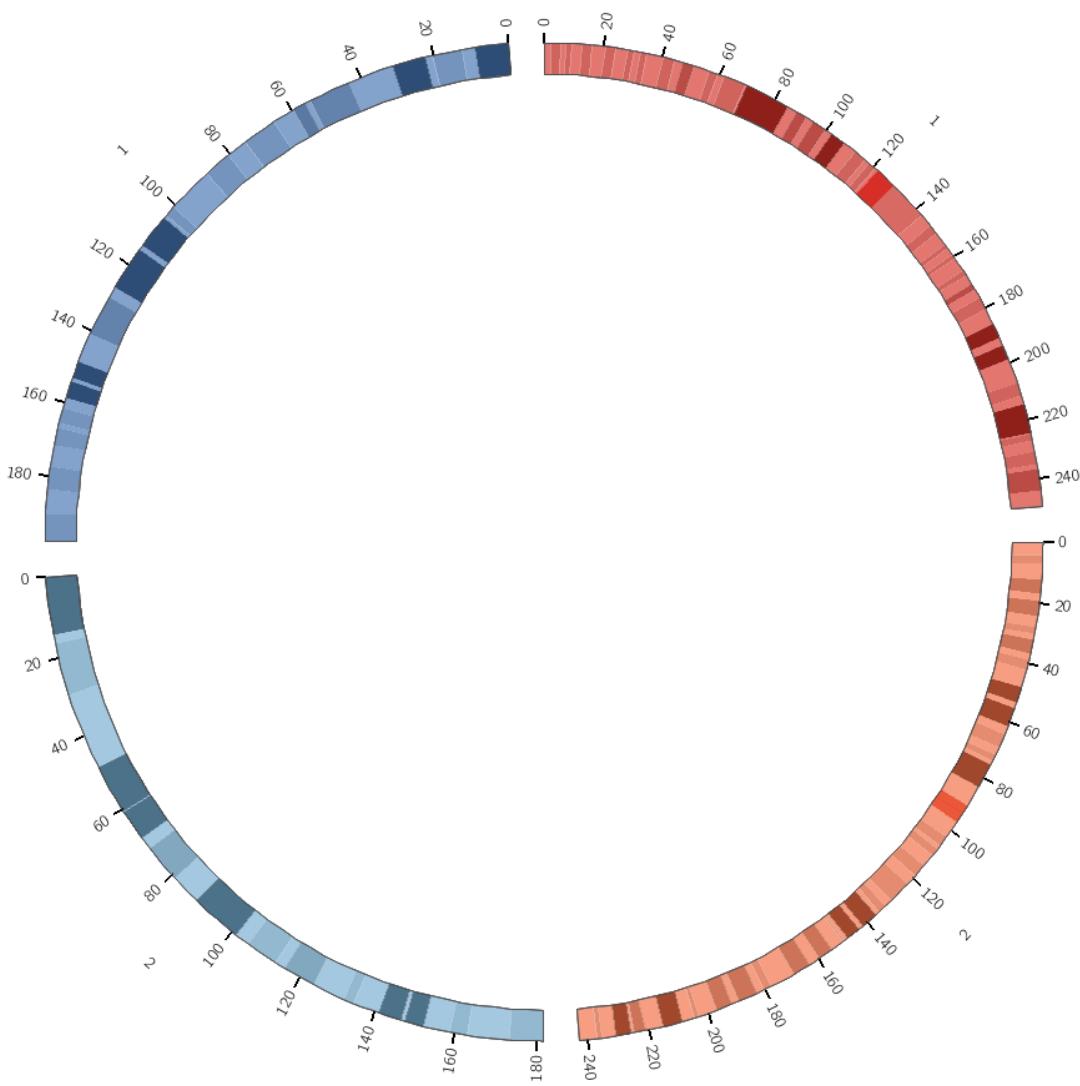


FIGURE 38

Chromosomes 1 and 2 of human and mouse genomes.

Scale of each ideogram has been adjusted so that it occupies 25% of the figure.

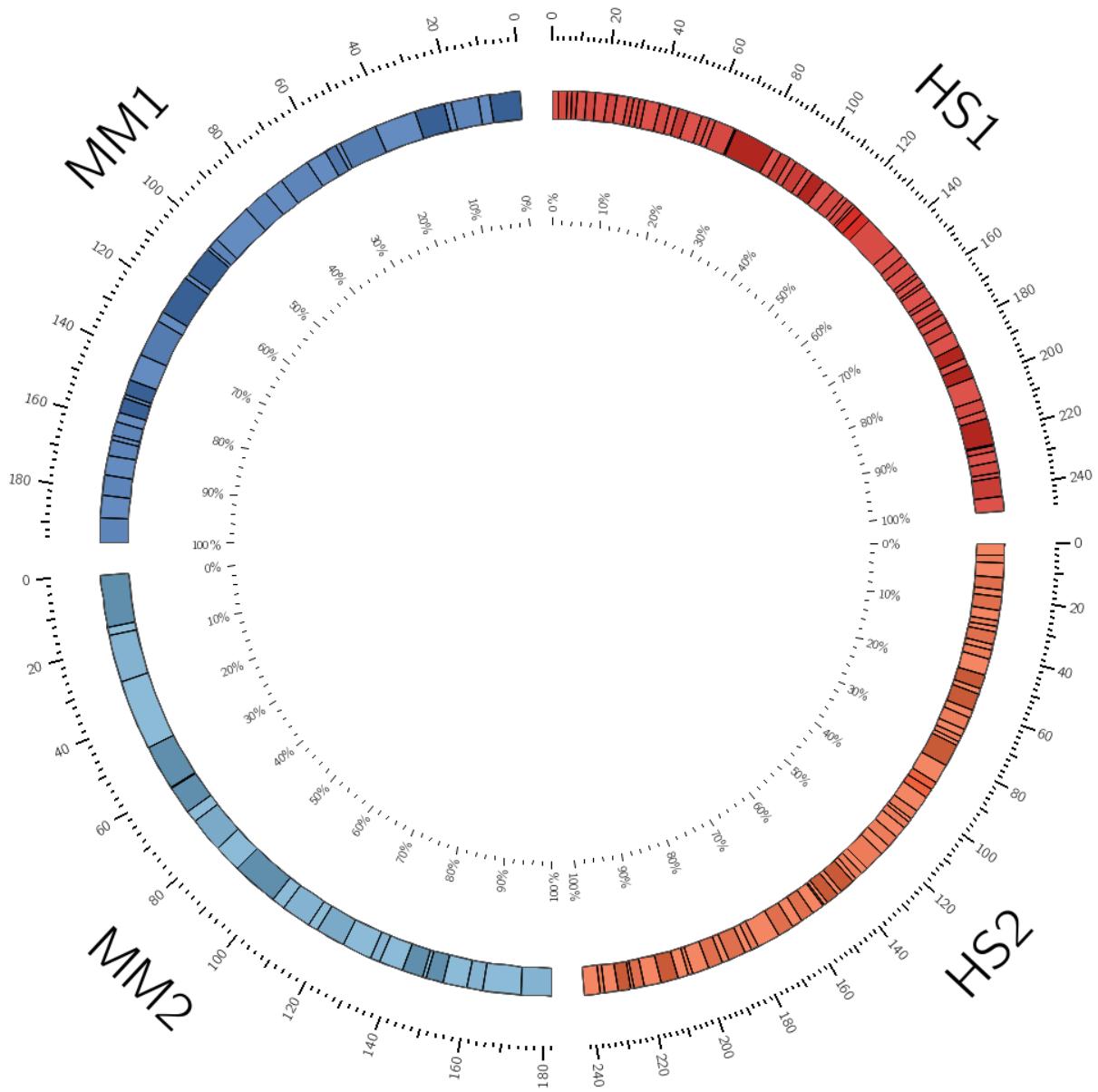


FIGURE 39

Chromosomes 1 and 2 of human and mouse genomes with multiple tick rings.

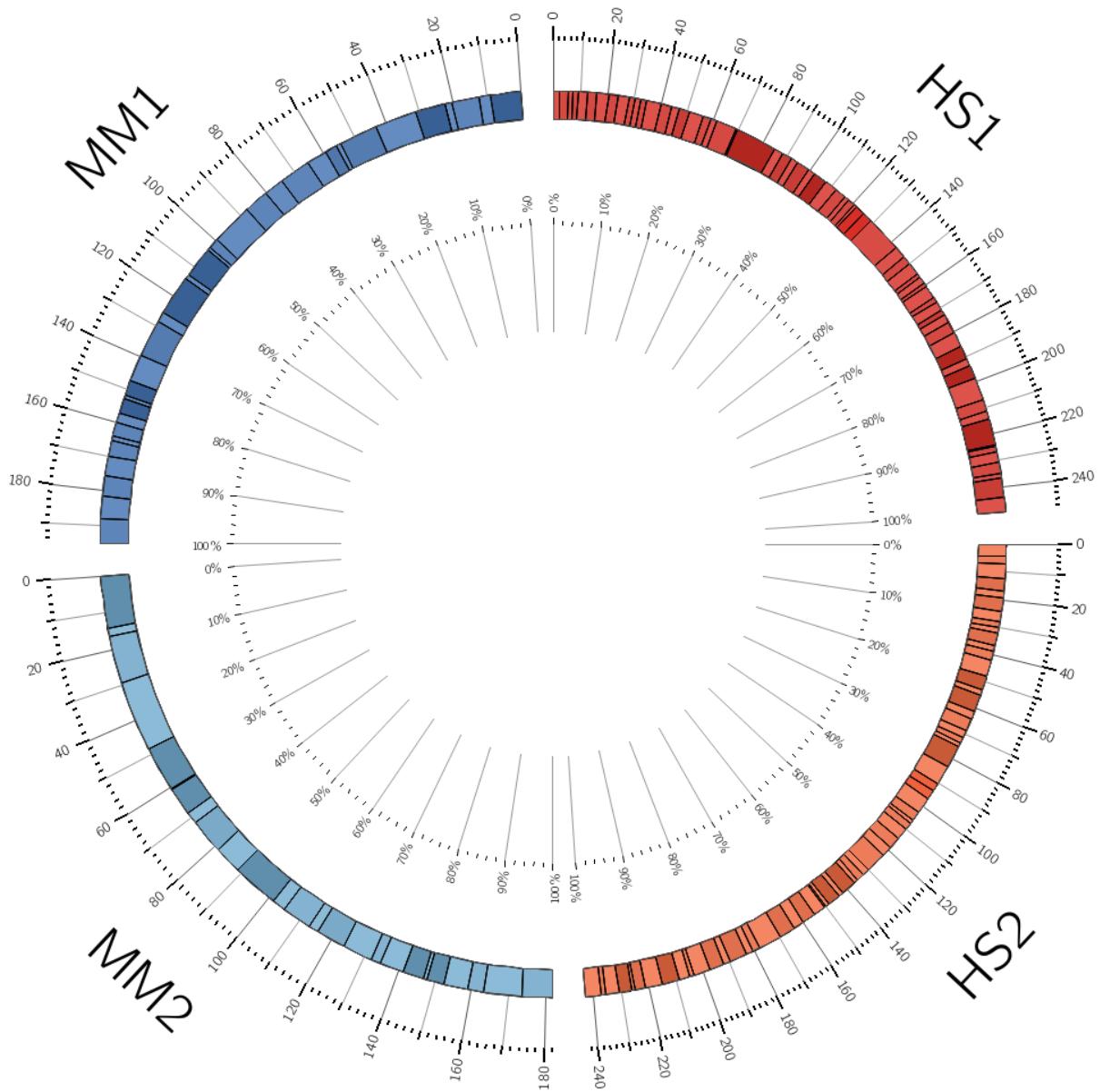


FIGURE 40

Chromosomes 1 and 2 of human and mouse genomes, as shown in Figure 39, but with grids.

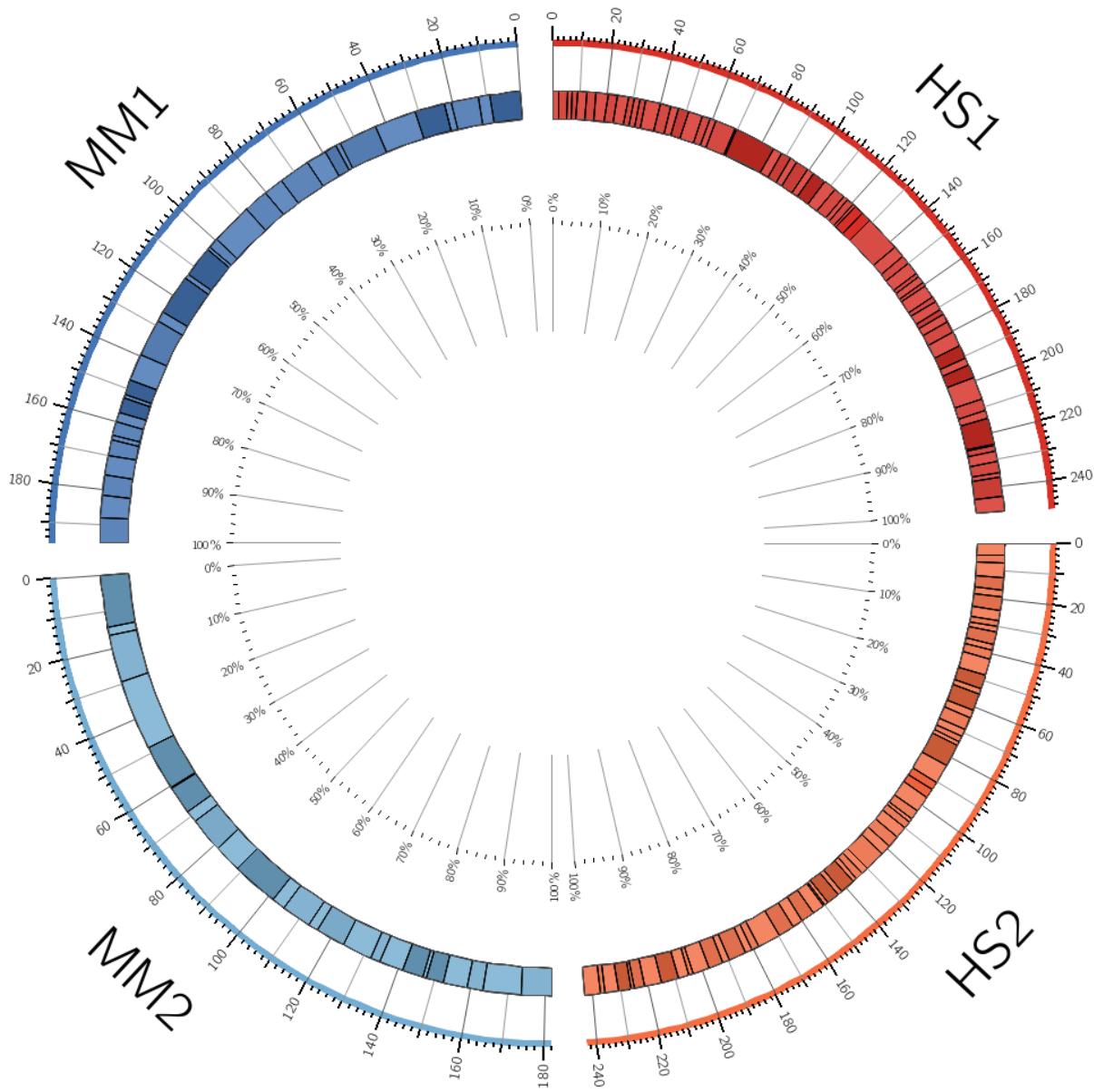


FIGURE 41

Chromosomes 1,2 of human and mouse genomes, as shown in Figure 40, but with highlight regions placed immediately inside the outer tick ring.