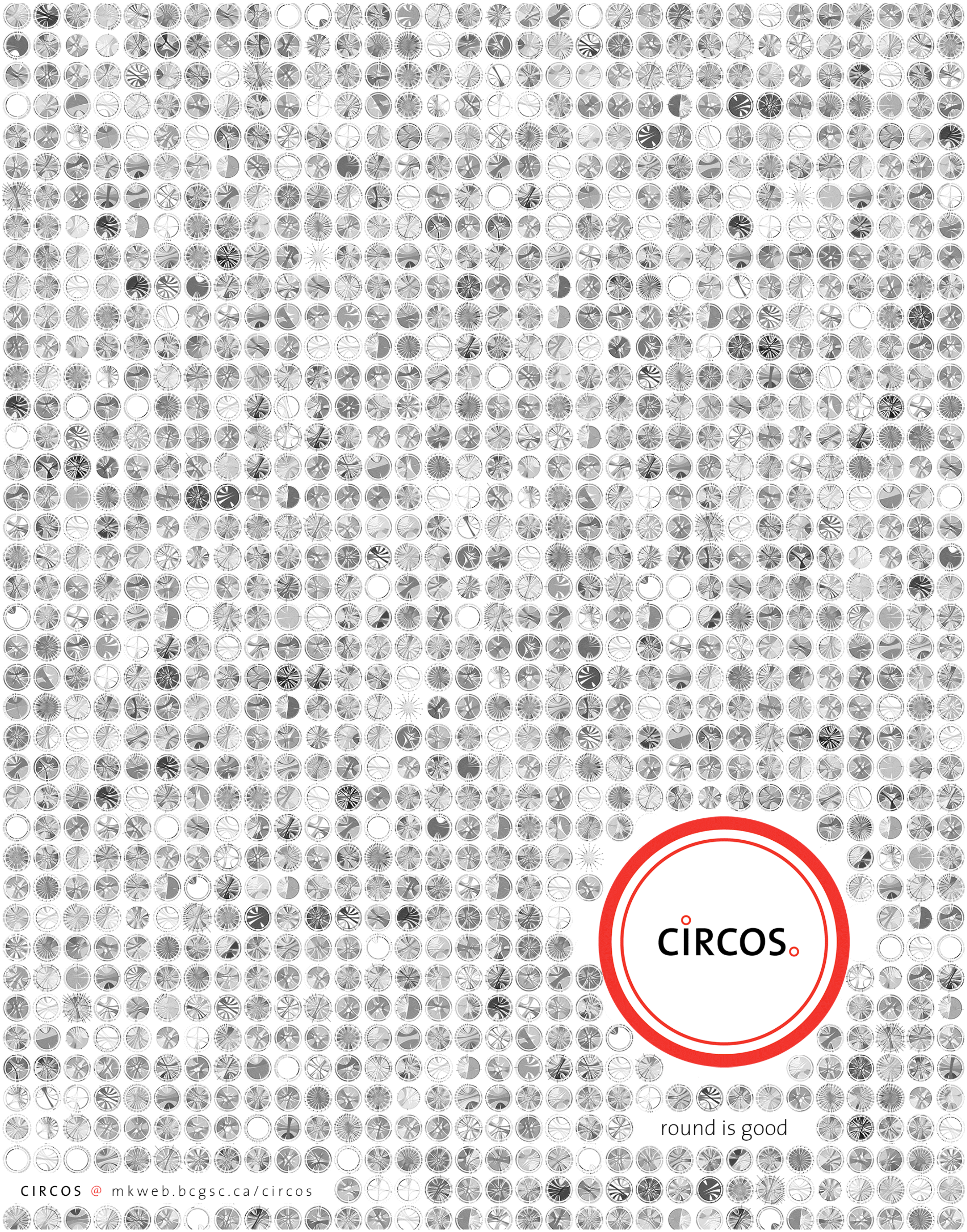
Version History

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

Introduction to Ideogram Layout

The legibility of the figure depends on an ideogram organization that is compatible with the data type and density. 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.

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

Each lesson has its own configuration file which defines parameters and files that are used to draw the image. For each lesson this file is `S/L/etc/circos.conf` where S is the session number (`S=2,3,4`) and `L` is the lesson number (`L=1,2,3 ...`). To keep this file tidy, content from other files will be imported using the `<<include>>` directive (this syntax is provided by the `Config::General` Perl module).

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.

```
# imports content from other configuration files, found
# in the same directory as the central configuration.

# 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 same directory as circos.conf
<<include ticks.conf>>

# size, format and location of the output file
<image>
file = s2-1.png
# from shared etc/ directory for Session 2 (sessions/2/etc)
<<include ../etc/image.conf>>
</image>

# definition of the karyotype, which defines the
# name, size and color of chromosomes
karyotype = ../data/karyotype.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

# the housekeeping file contains settings and imports
# that do not change between sessions and lessons (e.g. color
# definitions, font definitions, etc).
# from sessions/etc
<<include ../../etc/housekeeping.conf>>
```

For this lesson, let's focus on the karyotype file (in **bold** above), 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.

The colors used in this file must be defined in the `<colors>` block in the configuration file (this block is defined in the `housekeeping.conf` file, which is imported). A large number of colors are predefined for each hue (e.g. `vdred`, `dred`, `red`, `lred`, `vlred`, `v=very`, `d=dark`, `l=light`) as well as Brewer palettes (see *colorbrewer.org*). The 5-color qualitative pastel palette (`bq1-1 bq1-2 bq1-3 bq1-4 bq1-5`) is shown in Figure 1.

The karyotype file for this lesson is found in `sessions/2/data/karyotype.txt`.

```
# sessions/2/data/karyotype.txt

# first two fields (chr -) are required, followed by
# chromosome name (chr2)
# chromosome label (2)
# chromosome start (0)
# chromosome end (5000000
# chromosome color (bq1-1)
chr - chr1 1 0   5000000 bq1-1
chr - chr2 2 0  10000000 bq1-2
chr - chr3 3 0  20000000 bq1-3
chr - chr4 4 0  50000000 bq1-4
chr - chr5 5 0 100000000 bq1-5
```

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

```
> cd sessions/2/1
> circos –conf etc/circos.conf
...
created image at ./s2-1.png
```

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 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

```
# sessions/2/1/etc/ideogram.conf
<ideogram>
<spacing>
default = 2u
#default = 10u
#<pairwise chr1;chr2>
#spacing = 2u
#</pairwise>
#<pairwise chr5;chr1>
#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 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.

```
# sessions/2/1/etc/ideogram.conf
<ideogram>
<spacing>
#default = 2u
default = 10u
<pairwise chr1;chr2>
spacing = 2u
</pairwise>
#<pairwise chr5;chr1>
#spacing = 25u
#</pairwise>
</spacing>
...
</ideogram>
```

By uncommenting the first `<pairwise>` block, the space between chromosomes 1 and 2 is changed from `10u` (which is the new default) to `2u`. The adjusted spacing is shown in Figure 4 – note that all ideograms are spaced at `10u`, except chromosomes 1 and 2, which are spaced by `2u`.

You can define as many `<pairwise>` blocks as you need. By commenting out the second `<pairwise>` block, the spacing between chromosomes 5 and 1 is increased to `25u`. The result is shown in Figure 5.

```
# sessions/2/1/etc/ideogram.conf
<ideogram>
<spacing>
#default = 2u
default = 10u
<pairwise chr1;chr2>
spacing = 2u
</pairwise>
<pairwise chr5;chr1>
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 the `default = 0.1r` line, spacing is set to 18.5 Mb (10% of total ideogram size, which is 185 Mb). The result is shown in Figure 6.

```
#sessions/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 `2u`, respectively. 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 = 1u # 1Mb space
</pairwise>

<pairwise chr3;chr4>
spacing = 2u # 2Mb space
</pairwise>

</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 5% of the ideogram circle's circumference to spacing (if so, use `0.05r`). 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 *10x* magnification), or if you wish to reduce the importance of an ideogram (e.g. show chromosome 2 at *5x* 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 *0.2x* magnification (*5x* reduction), chromosome 2 at *2x* magnification and chromosome 3 at *10x* 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 *0.5x* magnification (*2x* reduction).

```
<<include ideogram.conf>>
<<include ticks.conf>>

<image>
file = s2-3.png
<<include ../etc/image.conf>>
</image>

karyotype = ../data/karyotype.txt

chromosomes_units           = 1000000
chromosomes_display_default = yes

chromosomes_scale = chr1:0.5
#chromosomes_scale = chr1:0.5;chr2:2;chr3:10
# chr1 occupies 50% of figure
#chromosomes_scale = chr1:38

# chr1 occupies 75% of figure
#chromosomes_scale = chr1:114

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

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 *1x*), chromosome 1 appeared to be ½ of the size of chromosome 2 (because its length was ½ of chromosome 2). Now that we have changed the scale of chromosome 1 to *0.5x* it appears to be ¼ 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
# chr1 occupies 50% of figure
#chromosomes_scale = chr1:38
```

in addition to shrinking chromosome 1, the size of chromosomes 2 and 3 will be changed to *2x* and *10x* original size, respectively. See Figure 9 for the results.

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, you need to change the scale, *s*, of chromosome 1. If we let the length of chromosome 1 be L_1, requiring that it occupy 50% of the figure requires that

$$\frac{s \cdot L_1}{s \cdot L_1 + (G - L_1) + S} = 0.5$$

where G is the total size of all ideograms displayed and S is the total size of all spaces between ideograms. In this example, $S = 10$ Mb (5 spaces, `2u` each), $L = 180$ Mb (recall that chromosome 1 is 5 Mb and all chromosomes total 185 Mb) and $L_1 = 5$ Mb.

$$\frac{5s}{(5s+180+10)} = 0.5$$

Solving gives $s = 38$. Thus, setting `chromosomes_scale` to `chr1:38`,

```
#chromosomes_scale = chr1:0.5
#chromosomes_scale = chr1:0.5;chr2:2;chr3:10
# chr1 occupies 50% of figure
chromosomes_scale = chr1:38

# chr1 occupies 75% of figure
#chromosomes_scale = chr1:114
```

yields the required result in Figure 10.

To make chromosome 1 occupy 75% of the figure, we need to solve

$$\frac{5s}{(5s+180+10)} = 0.75$$

which gives $s = 114$. Uncommenting the line

```
# chr1 occupies 75% of figure
chromosomes_scale = chr1:114
```

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.

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. For more details, see the online tutorial about scale.

http://mkweb.bcgsc.ca/circos/tutorials/lessons/scaling/

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 12. 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 13 shows the effect of using `0.01r` spacing on two figures with 35 Mb and 185 Mb of ideograms.

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.

```
chromosomes_units           = 1000000
chromosomes_display_default = yes

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

# relative order
#chromosomes_order = chr3,chr5

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

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

The change in order is shown in Figure 14.

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.

By setting chromosomes order to

```
# 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 15.

You can reorder several subsets of ideograms by separating the order of subsets by “-” in the `chromosomes_order` parameter. Figure 16 shows the effect of changing the order parameter to

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

In this case there are two subsets that are reordered: chromosomes 1,4 and chromosomes 3,5. In the first subset chromosome 1 acts as the anchor (it is placed first, as it would by default) and followed immediately by chromosome 4. The next chromosome to be drawn is chromosome 2, since it follows chromosome 1 by default. Any chromosome that does not appear in the order parameter is placed between anchored subsets in the order that it would normally appear. The next subset is anchored on chromosome 3, which appears after chromosome 2, as it normally would. Chromosome 3 is followed by chromosome 5, as requested by the order parameter.

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.

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*.

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 17.

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 18.

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. Figure 19 shows the result.

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` → 1Mb
`default = 0.1r` → 10 Mb	`break = 1u` → 1Mb
`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 Spacing & Breaks tutorial at

http://mkweb.bcgsc.ca/circos/tutorials/lessons/ideograms/

```
<ideogram>

<spacing>

default = 0.025r
# when relative, size of break is computed
# relative to default spacing
break   = 0.5r

axis_break_at_edge = yes
axis_break         = yes
axis_break_style   = 2

<break_style 1>
stroke_color = black
# fills the break with a solid color, which has a
# thickness defined below
fill_color   = blue
# thickness of the break fill, relative to
# ideogram thickness or absolute
thickness    = 0.25r
stroke_thickness = 2p
</break>

<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     = black
stroke_thickness = 3p
# relative to ideogram thickness or absolute
thickness        = 1.5r
</break>

...
```

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 20. 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[a]:0-11;chr4[b]:19-31
```

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

```
chromosomes_order = ^,chr1,a,chr3,b
```

You'll recognize the order shown above. New is the `^` character, which represents the start of the ideogram circle. Ideogram "`a`" (chr4:0-11 Mb) is displayed after chromosome 1. Next comes chromosome 3, followed by ideogram "`b`" (chr4:19-31 Mb). The result is shown in Figure 21.

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 22). 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
# the next two fields are the name and 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
band chr3 band8 band8 19000000 20000000 stalk
```

Figure 23 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.

You can adjust the transparency of the band pattern by using `band_transparency` within the `<ideogram>` block. 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 in the `color.conf` file. 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. Thus, `red_a1` is a nearly opaque version of red and `red_a5` is an almost transparent red. Figure 24 shows the effect of setting `band_transparency = 3`.

Lesson 10 – Drawing Multiple Genomes

Thus far, we've drawn figures of a sample genome (see `session/2/data/karyotype.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, mouse and rat are available in `session/2/data`.

`session/2/data/karyotype.human.txt` – human karyotype (GRCh37/hg19)

`session/2/data/karyotype.mouse.txt` – mouse karyotype (NCBI37/mm9) `session/2/data/karyotype.rat.txt` – rat karyotype (Baylor 3.4/rn4)

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, ...) and rat (*Rattus norvegicus*) are prefixed with "`rn`" (rn1, rn2, ...).

```
# session/2/data/karyotype.human.txt
chr - hs1 1 0 247249719 chr1
chr - hs2 2 0 242951149 chr2
chr - hs3 3 0 199501827 chr3
chr - hs4 4 0 191273063 chr4
chr - hs5 5 0 180857866 chr5
chr - hs6 6 0 170899992 chr6
chr - hs7 7 0 158821424 chr7
...
```

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

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

we obtain a figure of all the human chromosomes (Figure 25). When the karyotype.mouse.txt file is used, we have the mouse genome, shown in Figure 26.

The UCSC genome browser uses a color convention to index human chromosomes. These colors are shown in Figure 27 and are defined in the color definition file `color.conf`. 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
chr0  = 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 27.

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. You will later learn about perceptual qualities of color and discover how to fix the fact that the very bright yellow of conventional color for chromosome 10 attracts the most attention.

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 bq1-1
chr - chr2 2 0  10000000 bq1-2
chr - chr3 3 0  20000000 bq1-3
chr - chr4 4 0  50000000 bq1-4
chr - chr5 5 0 100000000 bq1-5
```

where `bq1-N` corresponded to colors from a 5-color qualitative Brewer palette (Figure 1).

```
bq1-1 = 251,180,174
bq1-2 = 179,205,227
bq1-3 = 204,235,197
bq1-4 = 222,203,228
bq1-5 = 254,217,166
```

If you wish to display chromosomes from multiple genomes, it is sufficient to concatenate the corresponding karyotype files. Keep in mind that all chromosome names must be unique.

A karyotype file that combines both human and mouse genomes is available in `sessions/2/data/karyotype.human_mouse_labels.txt`. The chromosome labels in this file are defined as `hN` for human and `mN` for mouse chromosomes (see Figure 28). Make note to distinguish the *chromosome name* (which is the string used to reference the chromosome in the configuration file and data files) and the *chromosome label* (which is the string that appears in the image as the ideogram label).

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

If you are displaying only one genome it is sensible to make labels be the chromosome number. However, to distinguish ideograms from different species, labels should incorporate a unique species prefix (of your choosing) when displaying ideograms from different genomes.

Lesson 11 – Ideogram Progression and Orientation

In this lesson we will develop the ideogram layout used to show data, covered in subsequent lessons.

```
chromosomes_display_default = no
chromosomes       = hs1;hs2;mm1;mm2
chromosomes_order = hs1;hs2;mm2;mm1
```

First, we'll display only human chromosomes 1,2 and mouse chromosomes 1,2. This is done by setting `chromosomes_display_default = no` (to avoid displaying all of the chromosomes defined in the karyotype file), identifying which chromosomes to show using the `chromosomes` parameter, and setting the order of the ideograms using `chromosomes_order`. The result is shown in Figure 29.

In some cases, it is useful to be able to reverse the orientation of specific ideograms. In particular, when displaying multiple genomes, it is more intuitive that genomes have their origin (and end point) at the same point in the figure. By reversing the mouse chromosomes shown in Figure 29, and orienting the scale counter-clockwise, the start of mouse chromosome 1 is adjacent to the start of human chromosome 1. Similarly, the end of mouse chromosome 2 is adjacent to the end of chromosome 2.

This scale reversal is achieved using the `chromosomes_reverse` parameter.

```
chromosomes_reverse = mm2;mm1
```

The result is shown in Figure 30.

Finally, let’s make each ideogram in Figure 30 occupy 25% of the circle. To do so, we’ll need to change the scale of three ideograms, keeping hs1 as the scale reference. To make each ideogram appear the same size, we need

$$L_{hs1} = s_{hs2} L_{hs2} = s_{mm1} L_{mm1} = s_{mm2} L_{mm2}$$

where s_{chr} is the scale parameter of a chromosome and L_{chr} is its length. The sizes of each of the displayed chromosomes can be found in the karyotype file.

```
chr - hs1 1 0 247249719 chr1
chr - hs2 2 0 242951149 chr2
chr - mm1 1 0 197195432 white
chr - mm2 2 0 181748087 white
```

The scale for hs2, mm1 and mm2 is therefore

$$s_{hs2} = \frac{L_{hs1}}{L_{hs2}} = \frac{247.249}{242.95} = 1.018$$

$$s_{mm1} = \frac{L_{hs1}}{L_{mm1}} = \frac{247.249}{197.195} = 1.254$$

$$s_{mm2} = \frac{L_{hs1}}{L_{mm2}} = \frac{247.249}{181.748} = 1.360$$

The chromosomes_scale parameter is used to change the scale

```
chromosomes_scale   = hs2:1.018;mm1:1.254;mm2:1.360
```

The result is shown in Figure 31, where each ideogram occupies ¼ of the circle.

Lesson 12 – Relative and Absolute Ticks

In this final lesson of this session, we'll refine Figure 31 and prepare it for addition of data tracks. We will adjust the tick marks, chromosome labels and ideogram thickness.

First, we will change the thickness and position of the ideograms, as well as their label format. The parameters for these quantities are found in the `<ideogram>` block.

The parameters of concern here are `thickness` (set to `15p`, `p` = pixels, to make the ideograms thinner), `fill` (set to `no`, to avoid coloring the ideograms with the color specified in the karyotype file), and `radius` (set to `0.825r`, the radius of the ideogram circle, relative to the radius of the inscribed circle in the image).

The ideogram labels are also adjusted using the `label_*` parameters. The result is shown in Figure 32.

```
<ideogram>
...
thickness        = 15p
stroke_thickness = 1
# ideogram border color
stroke_color     = black
# ideogram is not filled with its color
fill             = no

# fractional radius position of chromosome ideogram within image
radius         = 0.825r
show_label     = yes
label_font     = condensedbold
label_radius   = dims(ideogram,radius) + 0.225r
label_size     = 24
# the labels will be parallel to the ideogram
label_parallel = yes
...
</ideogram>
```

In Figure 32 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.

```
#sessions/2/12/etc/ticks.conf

show_ticks          = yes
show_tick_labels    = yes

show_grid       = no

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

# absolute ticks (spacing is defined by parameter “spacing”)

<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>

<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>

<tick>
spacing        = 2u
size           = 3p
thickness      = 2p
show_label     = no
</tick>

# relative ticks (spacing is defined by parameter “rspacing”, with
spacing_type = relative)
```

```
<tick>
radius         = 0.75r
spacing_type   = relative
rspacing       = 0.02
size           = 3p
thickness      = 1p
show_label     = no
</tick>

<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 (at 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 32 with grids is shown in Figure 33.

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 `../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.

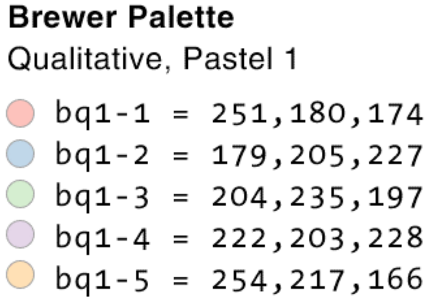
The contents of the `../data/highlight.txt` file are

```
# sessions/2/data/highlight.txt
# chr start end options
hs1 0 247249719 fill_color=bd2-1
hs2 0 242951149 fill_color=bd2-2
mm1 0 197195432 fill_color=bd2-4
mm2 0 181748087 fill_color=bd2-3
```

and define the highlight regions. 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 set to a diverging 4-color Brewer palette (defined in `colors.conf`).

The final figure, with highlights, is shown in Figure 34.

Figures



Brewer Palette

Qualitative, Pastel 1

```
● bq1-1 = 251,180,174
● bq1-2 = 179,205,227
● bq1-3 = 204,235,197
● bq1-4 = 222,203,228
● bq1-5 = 254,217,166
```

FIGURE 1

A 5-color qualitative Brewer palette.

For more palettes, see *colorbrewer.org.*

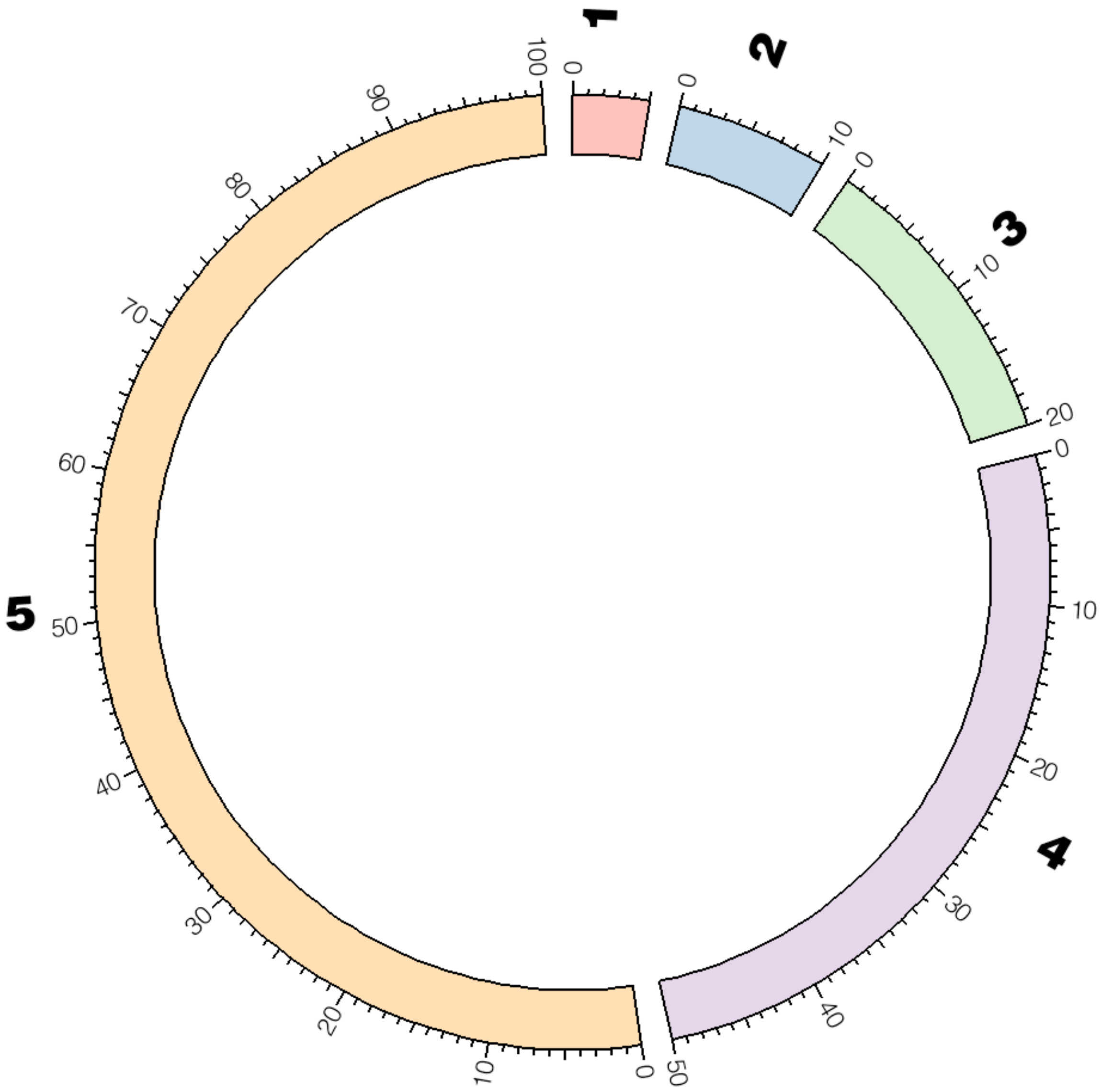


FIGURE 2

Five ideograms, arranged clockwise from 12 o’clock.

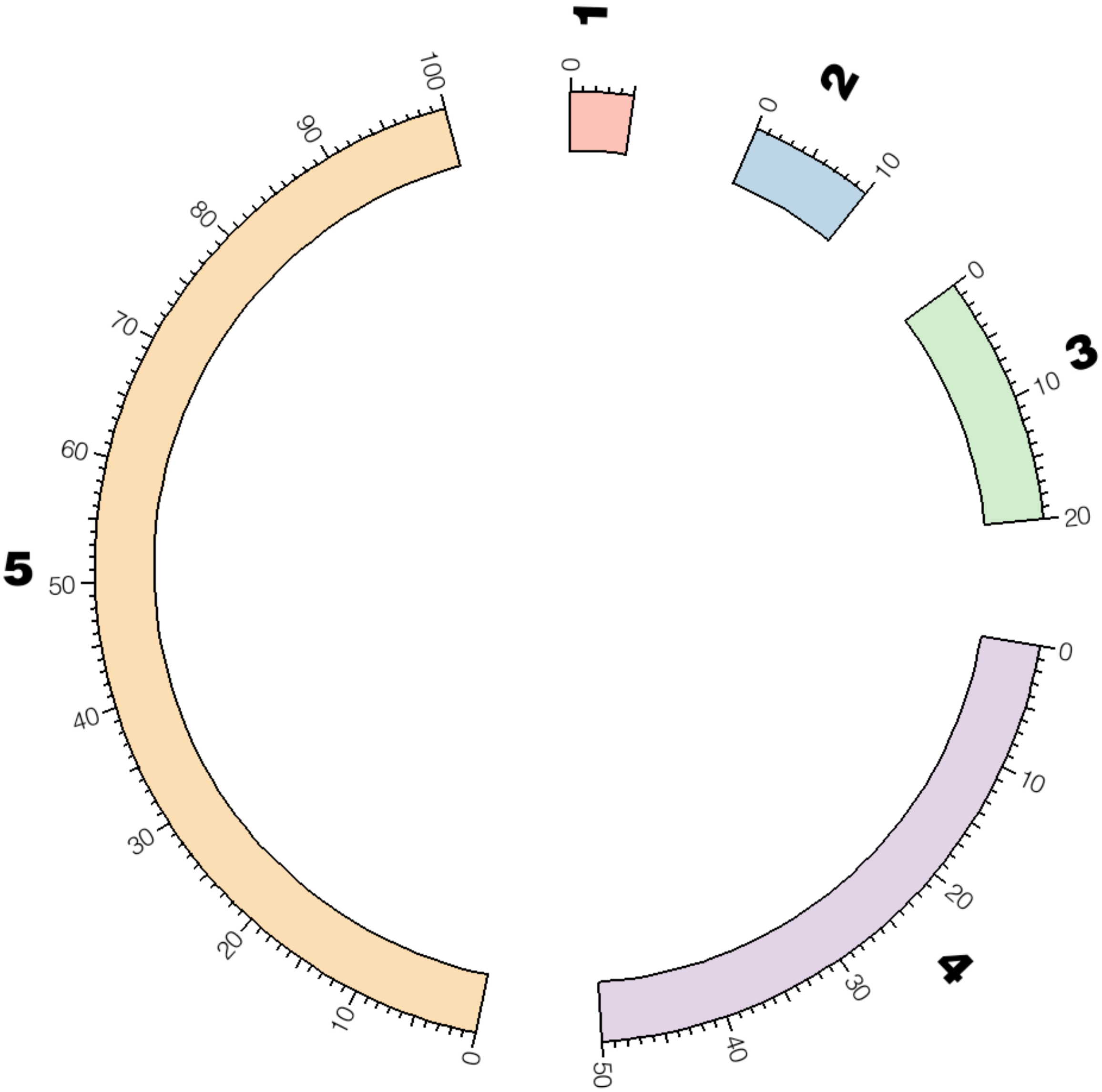


FIGURE 3

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

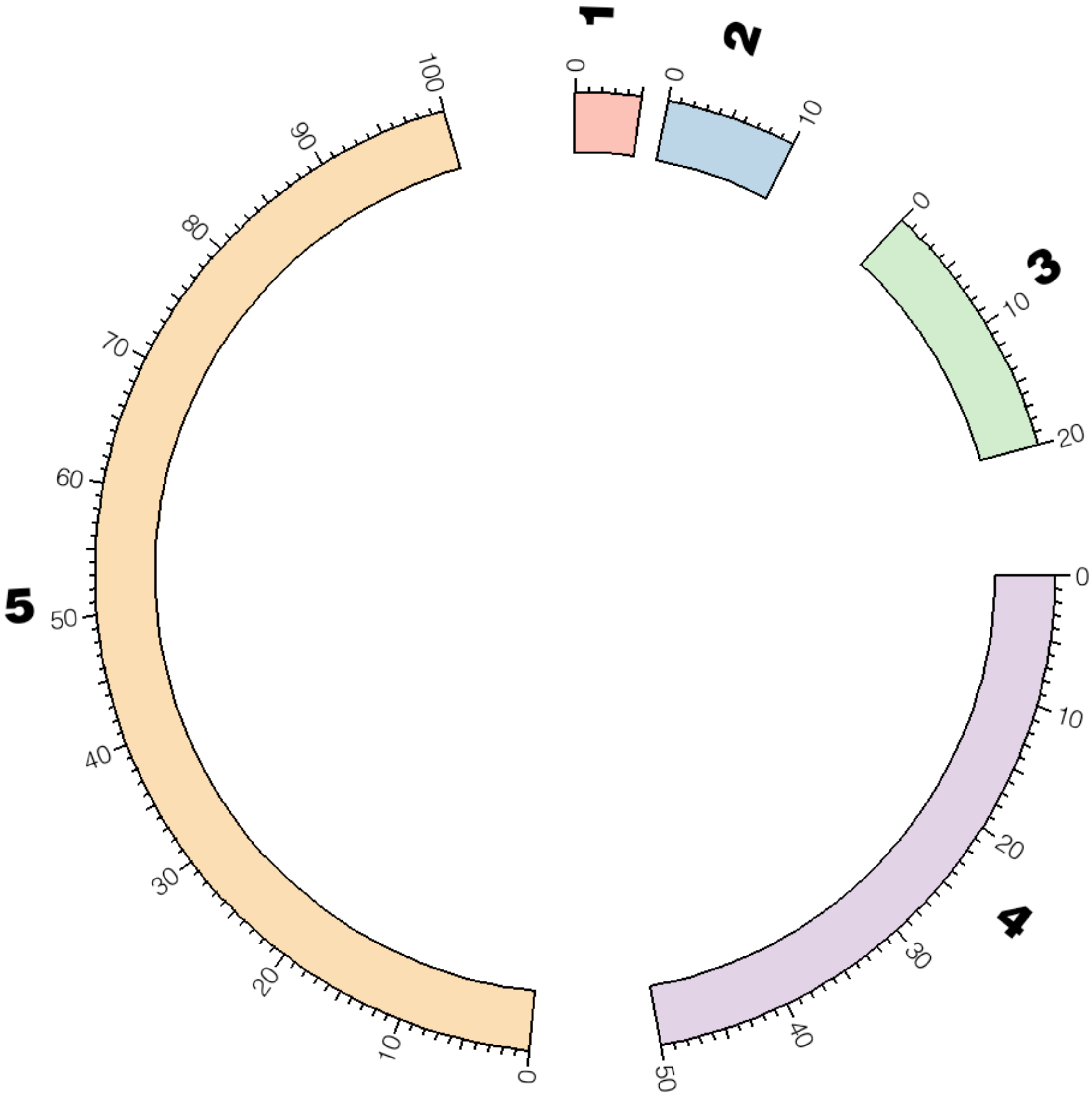


FIGURE 4

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

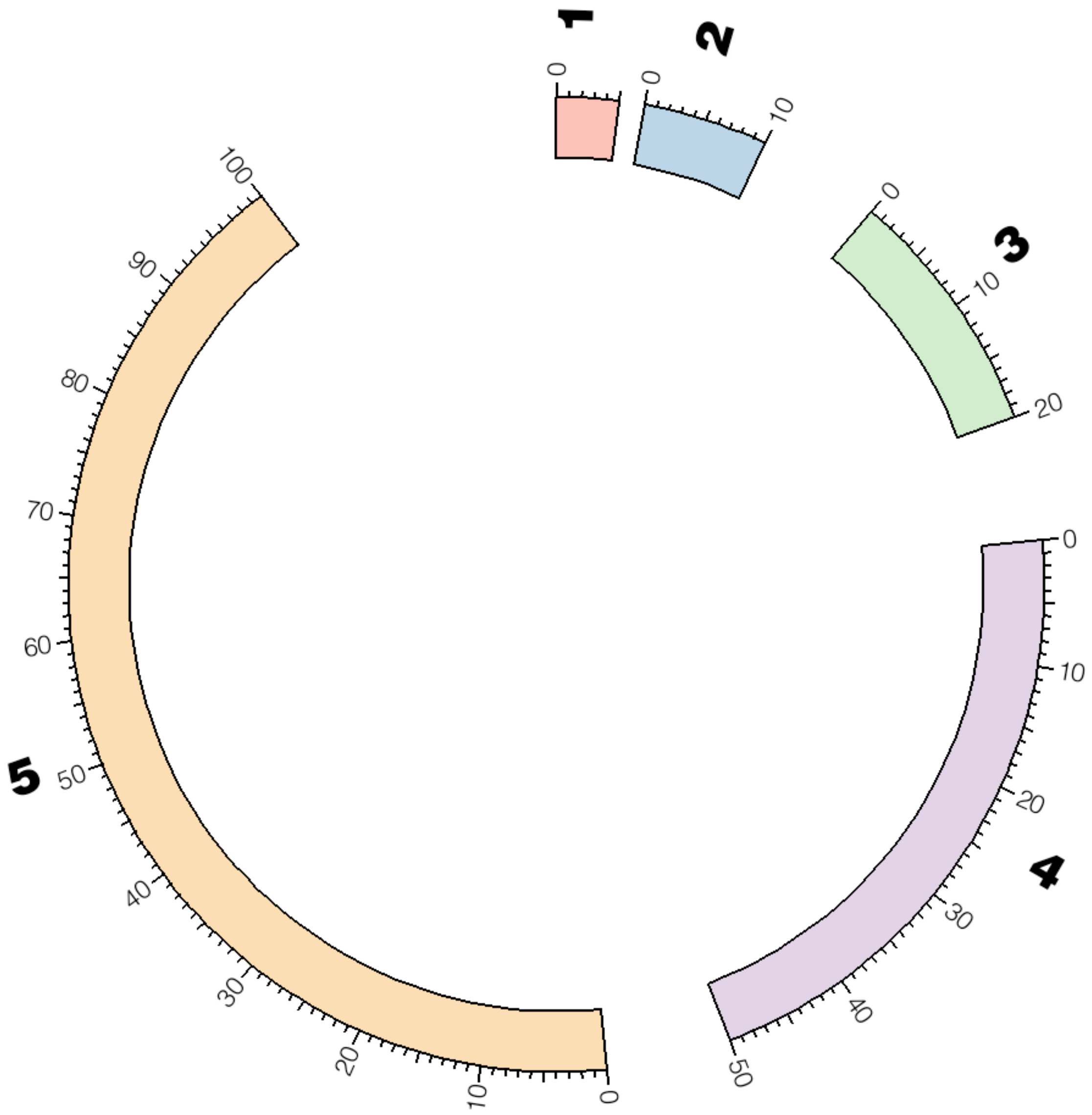


FIGURE 5

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

In this figure, two such blocks are used to decrease the spacing between chromosomes 1 and 2 to `2u` and increase the spacing between chromosomes 5 and 1 to `25u`.

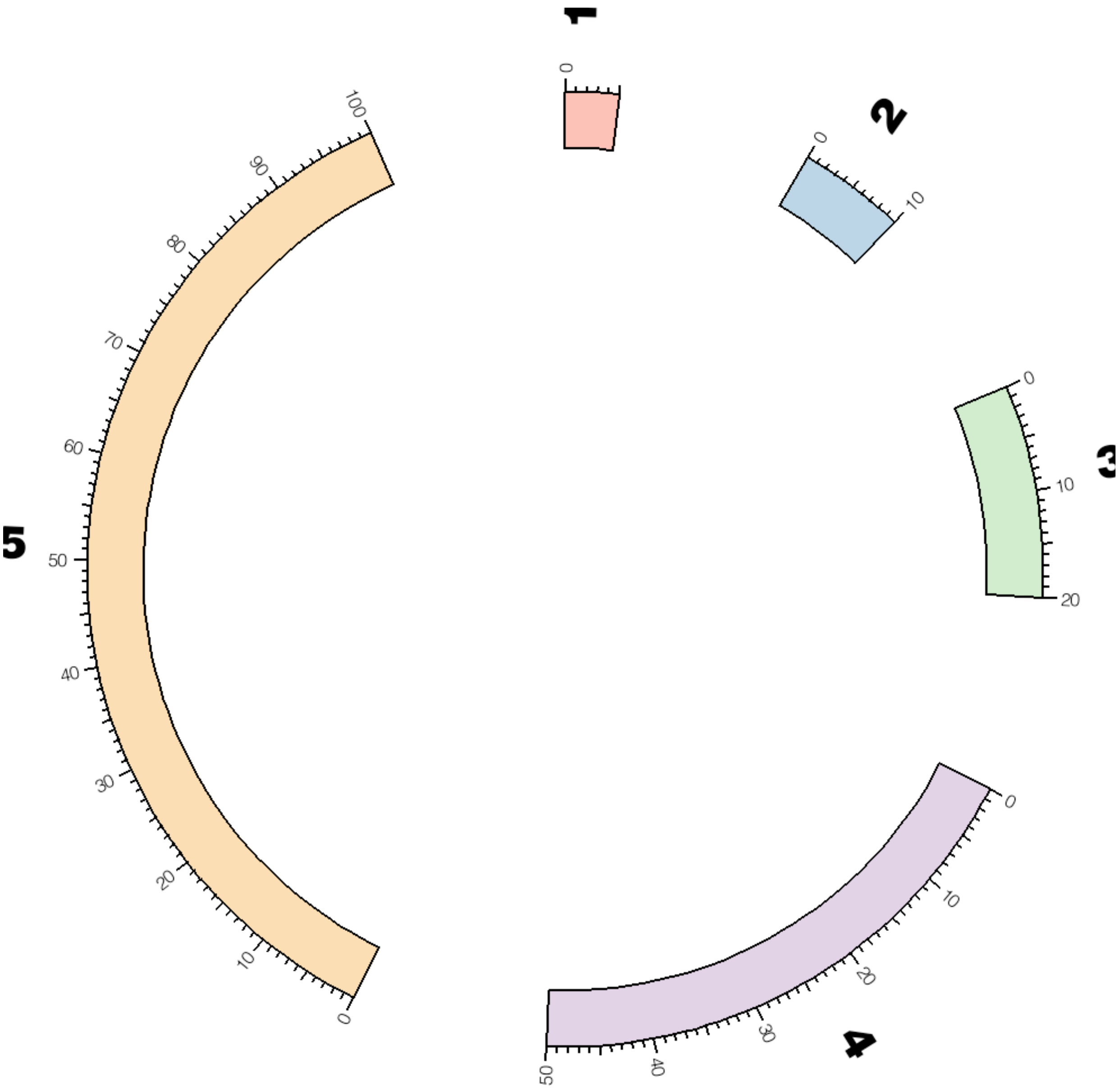


FIGURE 6

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>
```

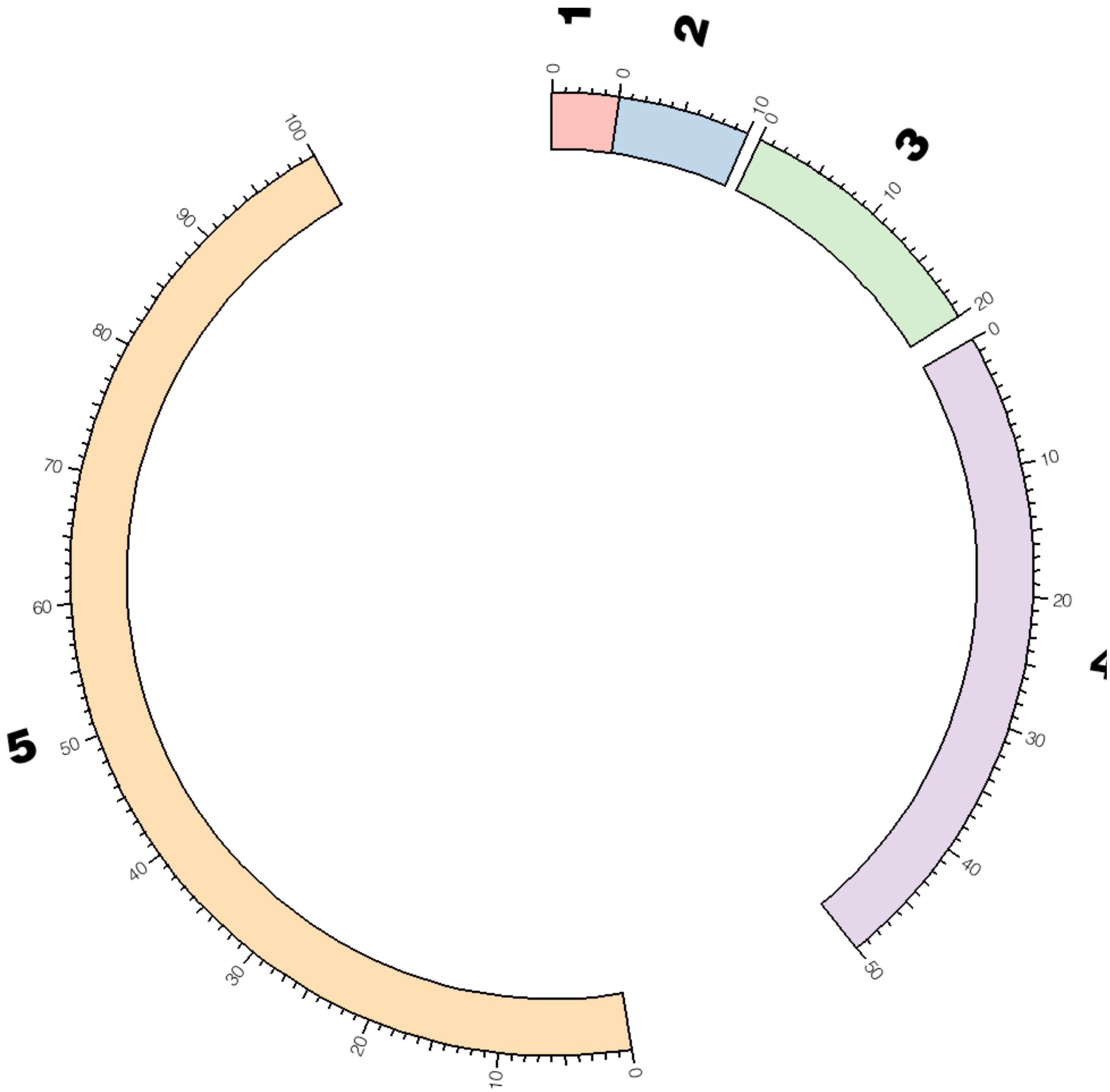


FIGURE 7

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`, `1u` and `2u`, respectively.

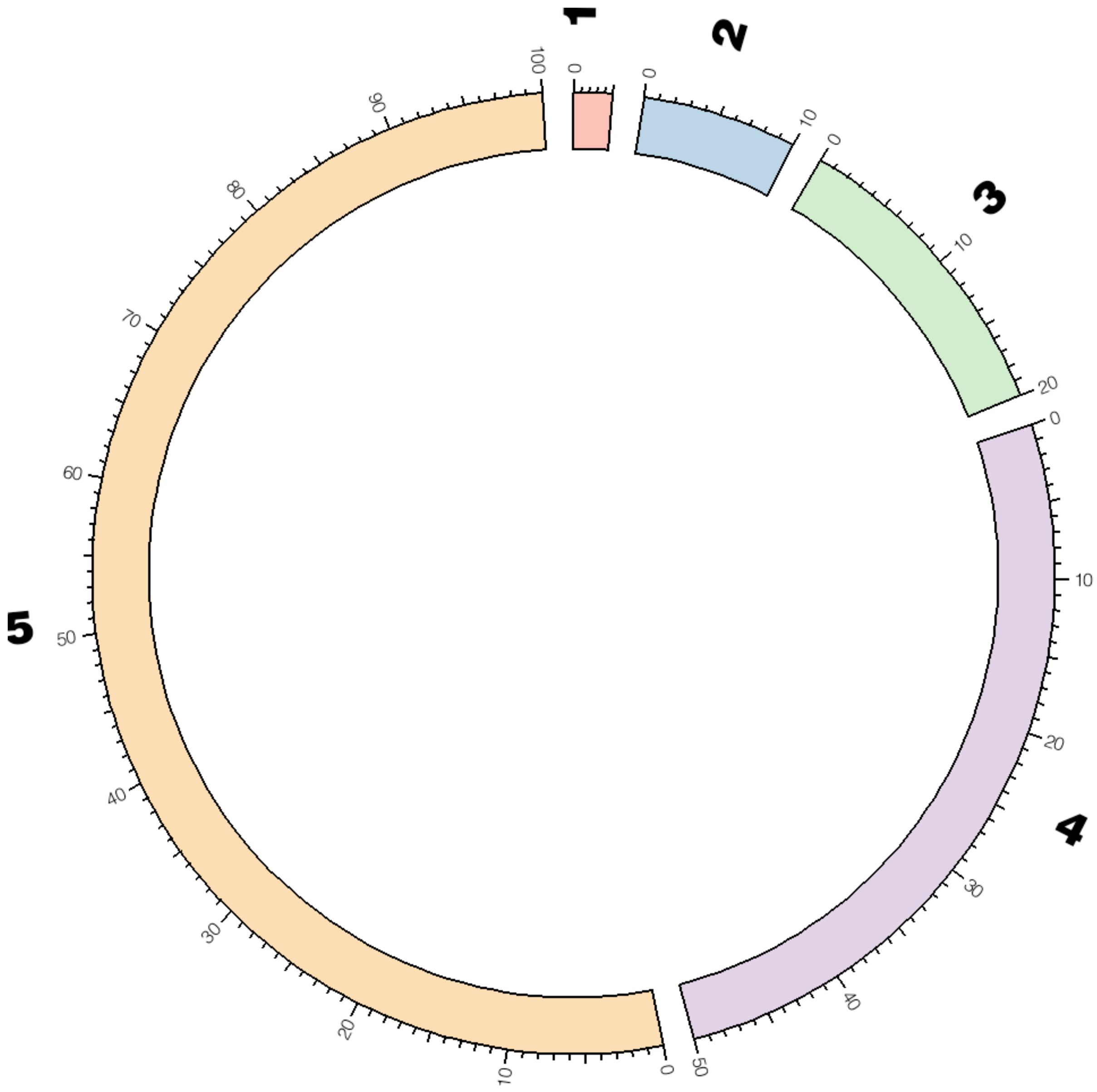


FIGURE 8

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

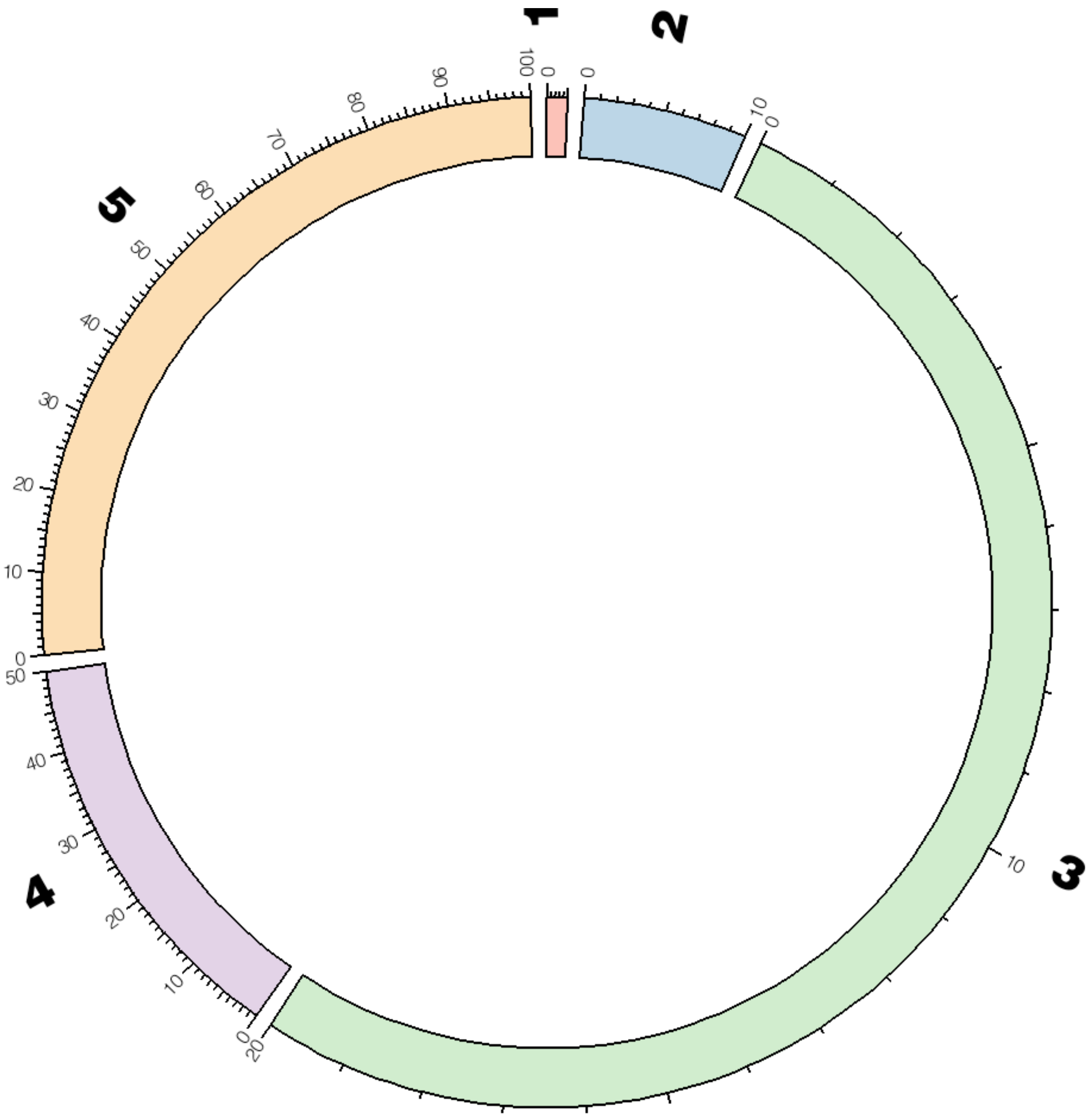


FIGURE 9

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).

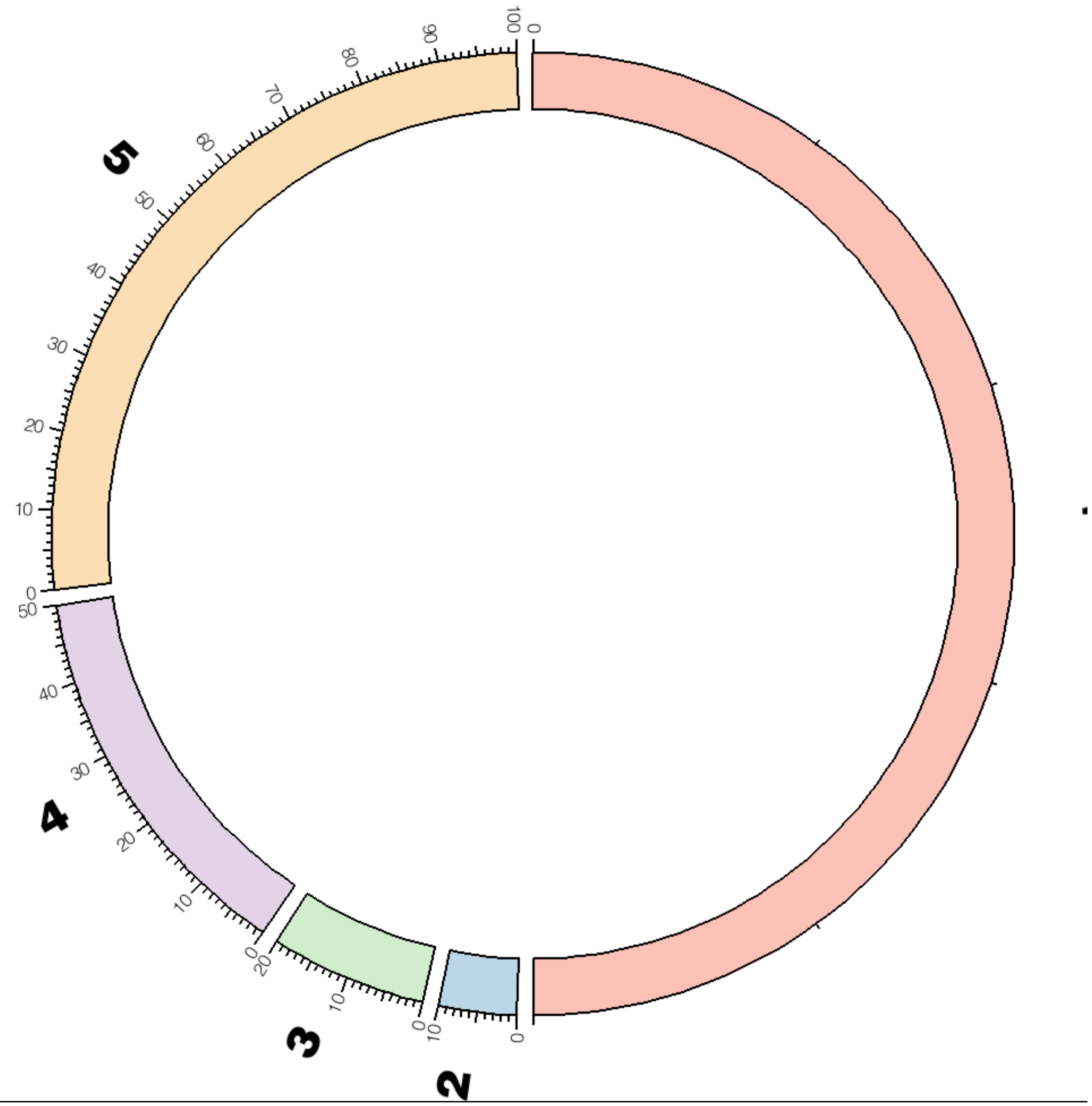


FIGURE 10

By changing the scale of chromosome 1 to 38, it is made to occupy 50% of the figure.

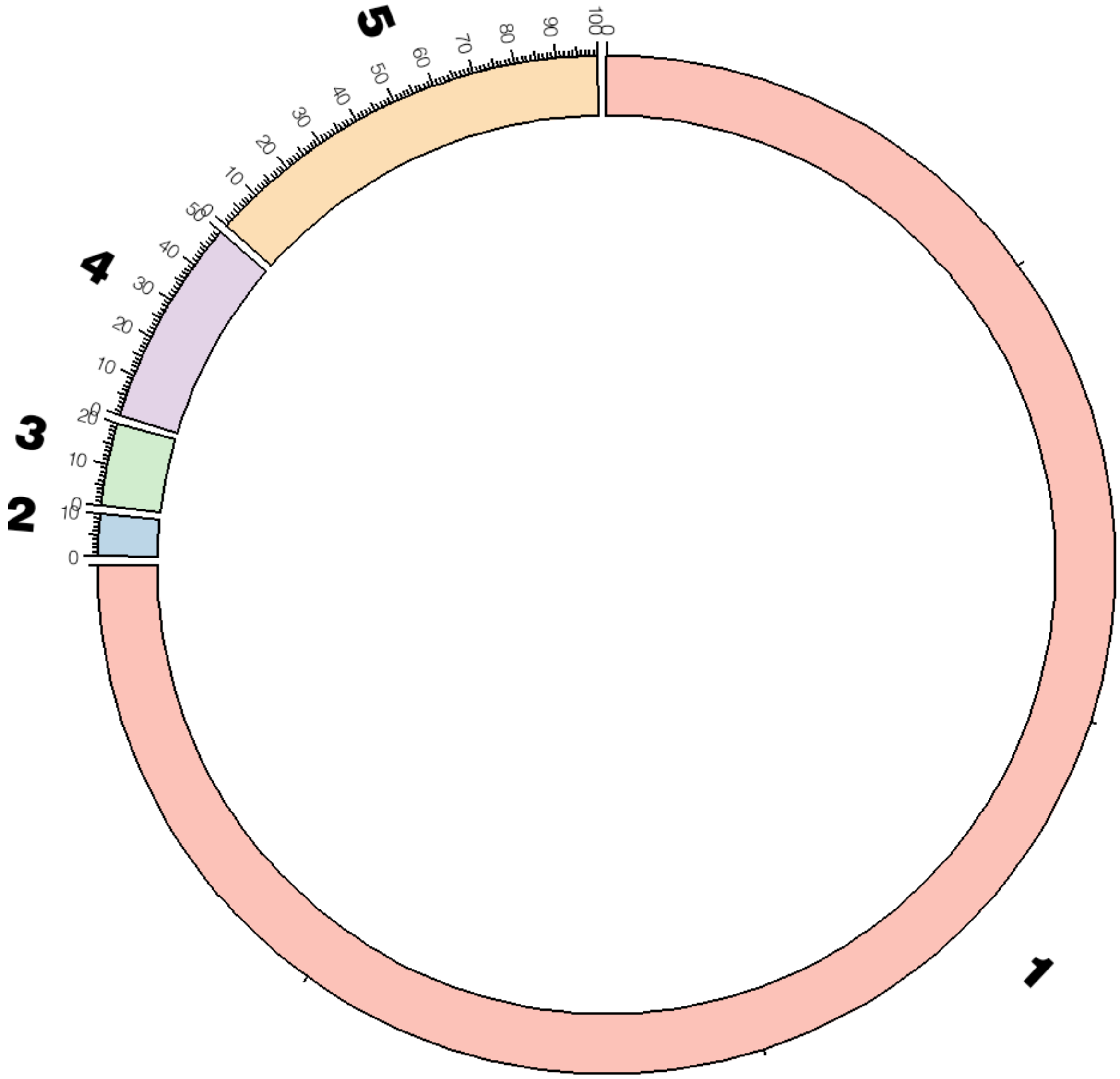


FIGURE 11

By changing the scale of chromosome 1 to 114, it is made to occupy 75% of the figure.

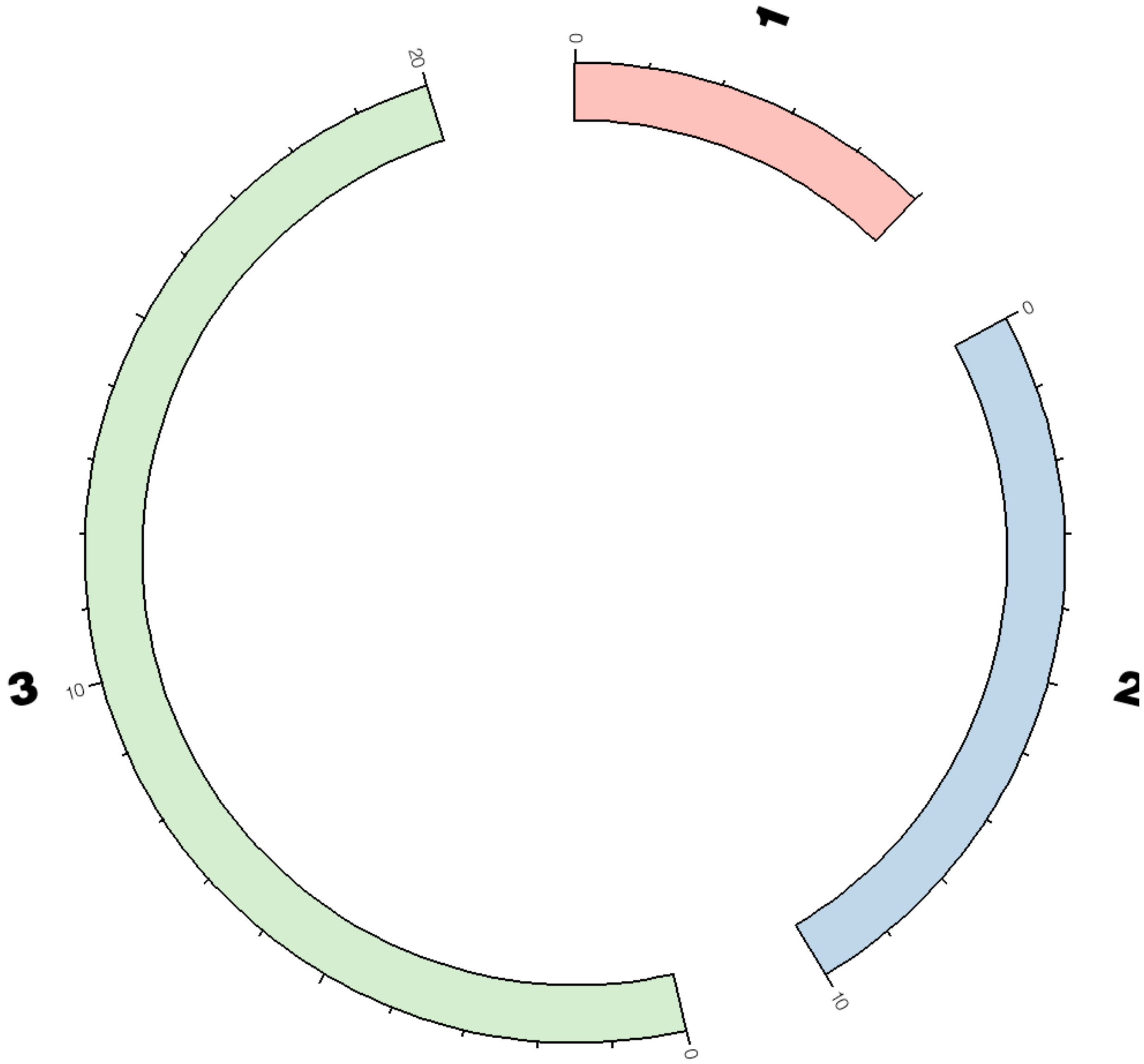


FIGURE 12

By turning off default chromosome display and setting the `chromosomes` parameter, you can control which chromosomes are drawn.

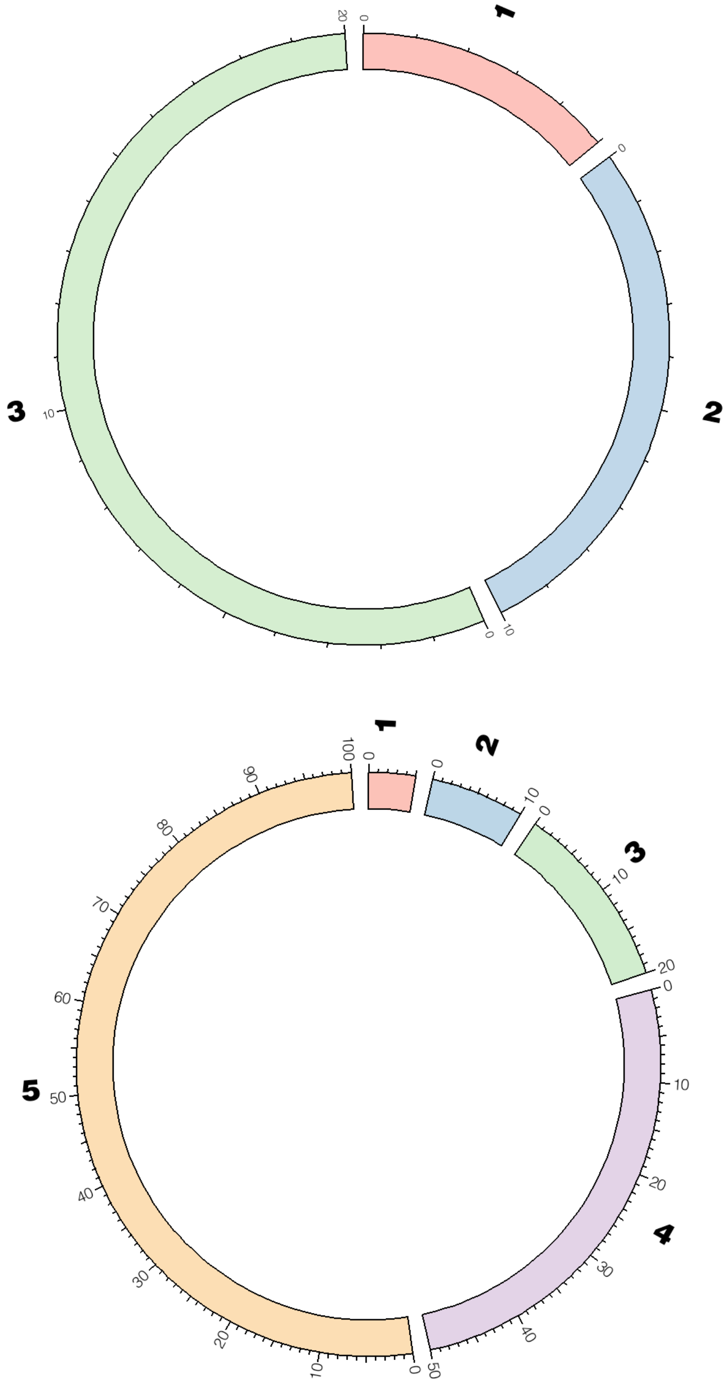


FIGURE 13

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, spacin gis set to `0.01r` and the total ideogram size is 35Mb. In the bottom panel, spacing is also `1u` but ideogram size is 185Mb.

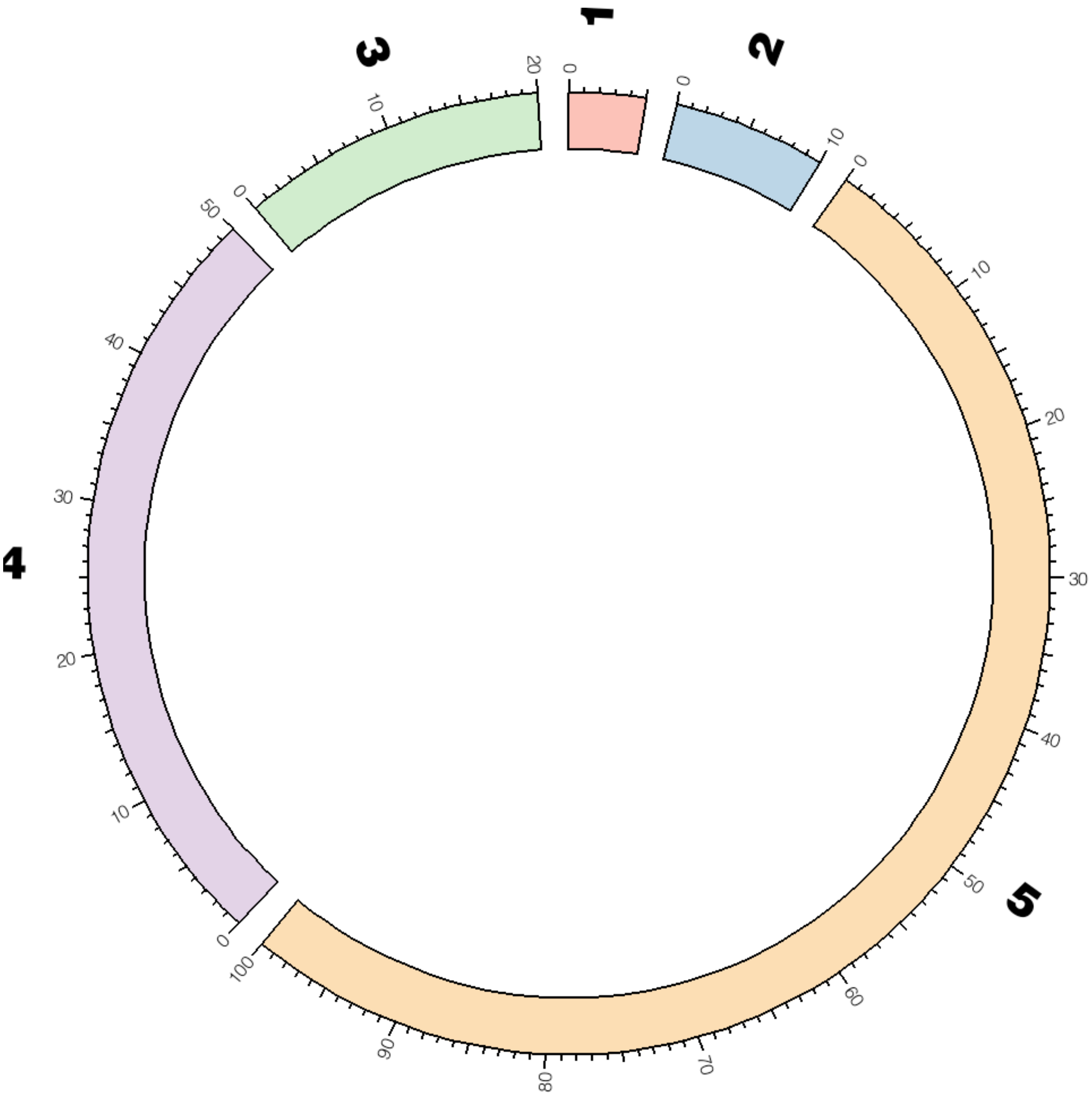


FIGURE 14

Chromosome order is adjusted using

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

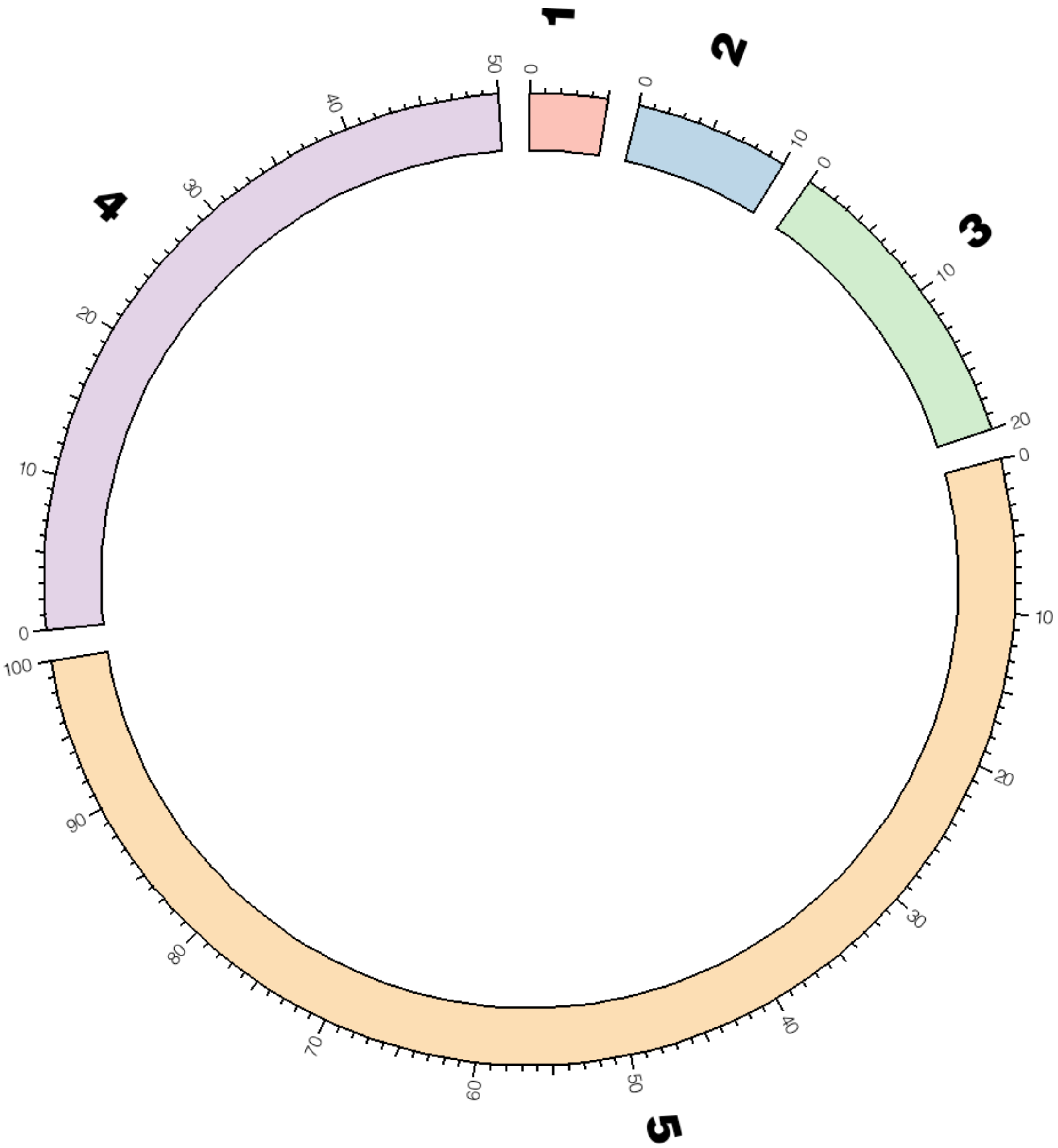


FIGURE 15

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
```

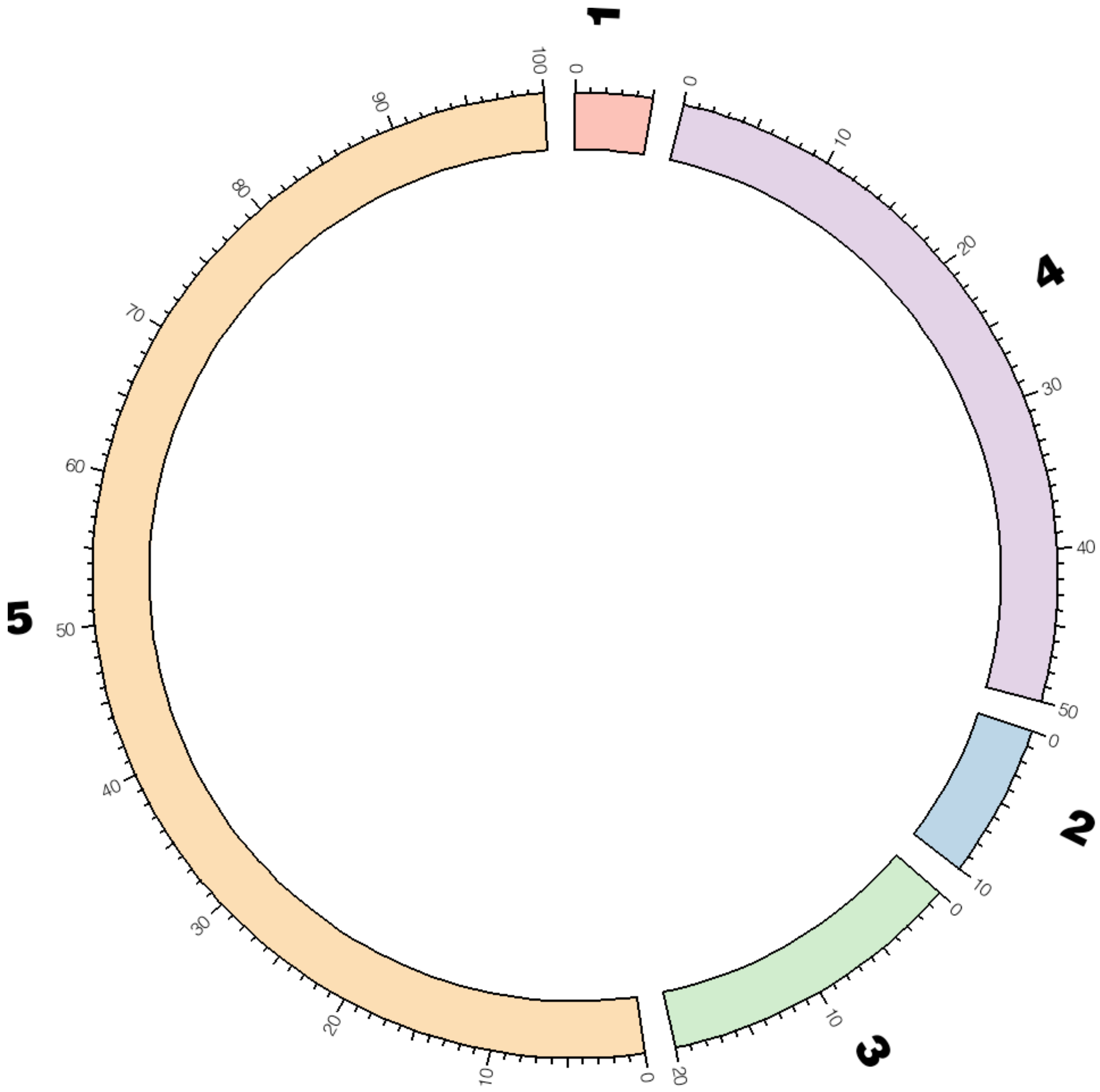


FIGURE 16

Chromosome order is adjusted using the `chromosomes_order` parameter.

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

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

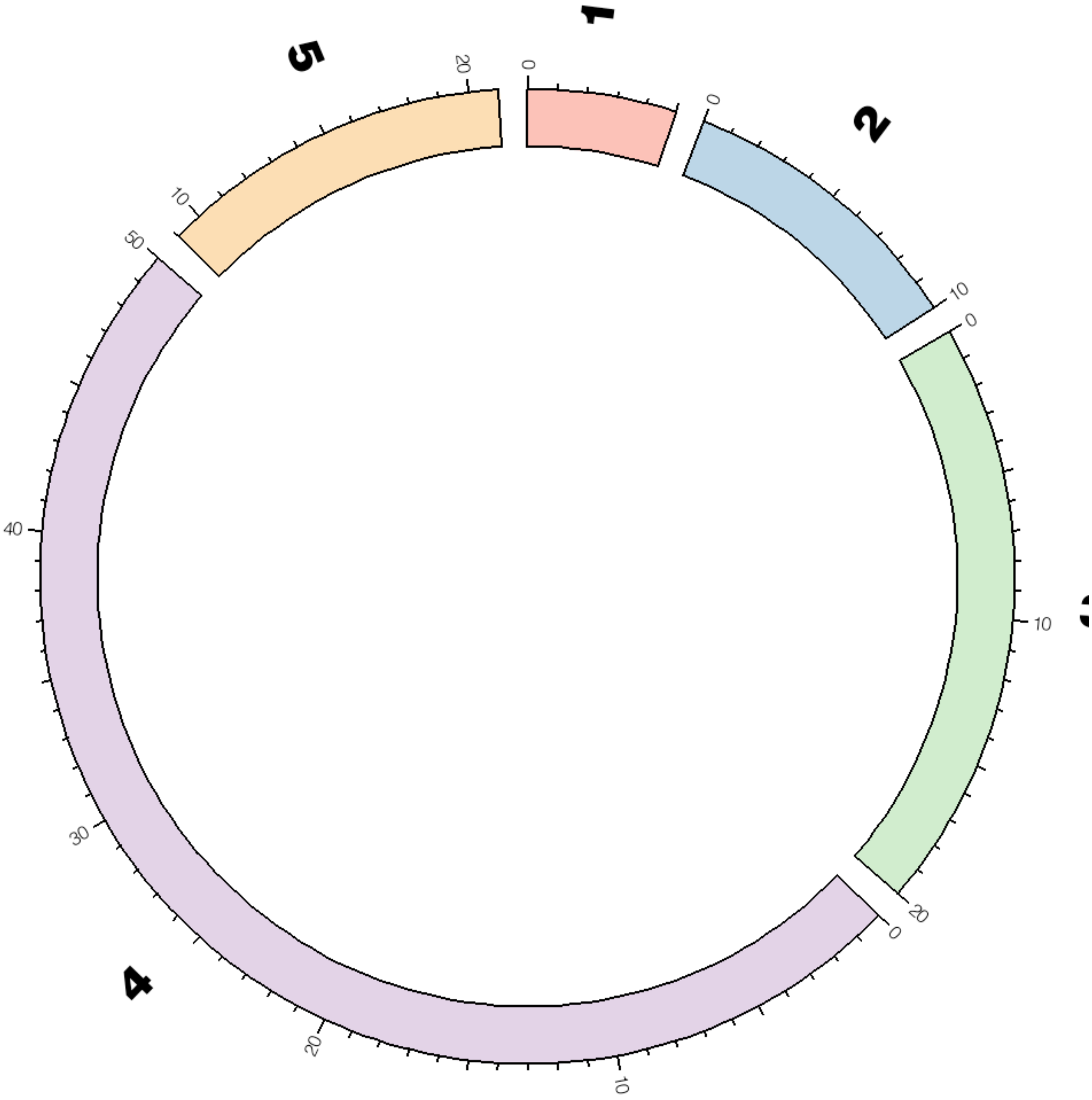


FIGURE 17

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.

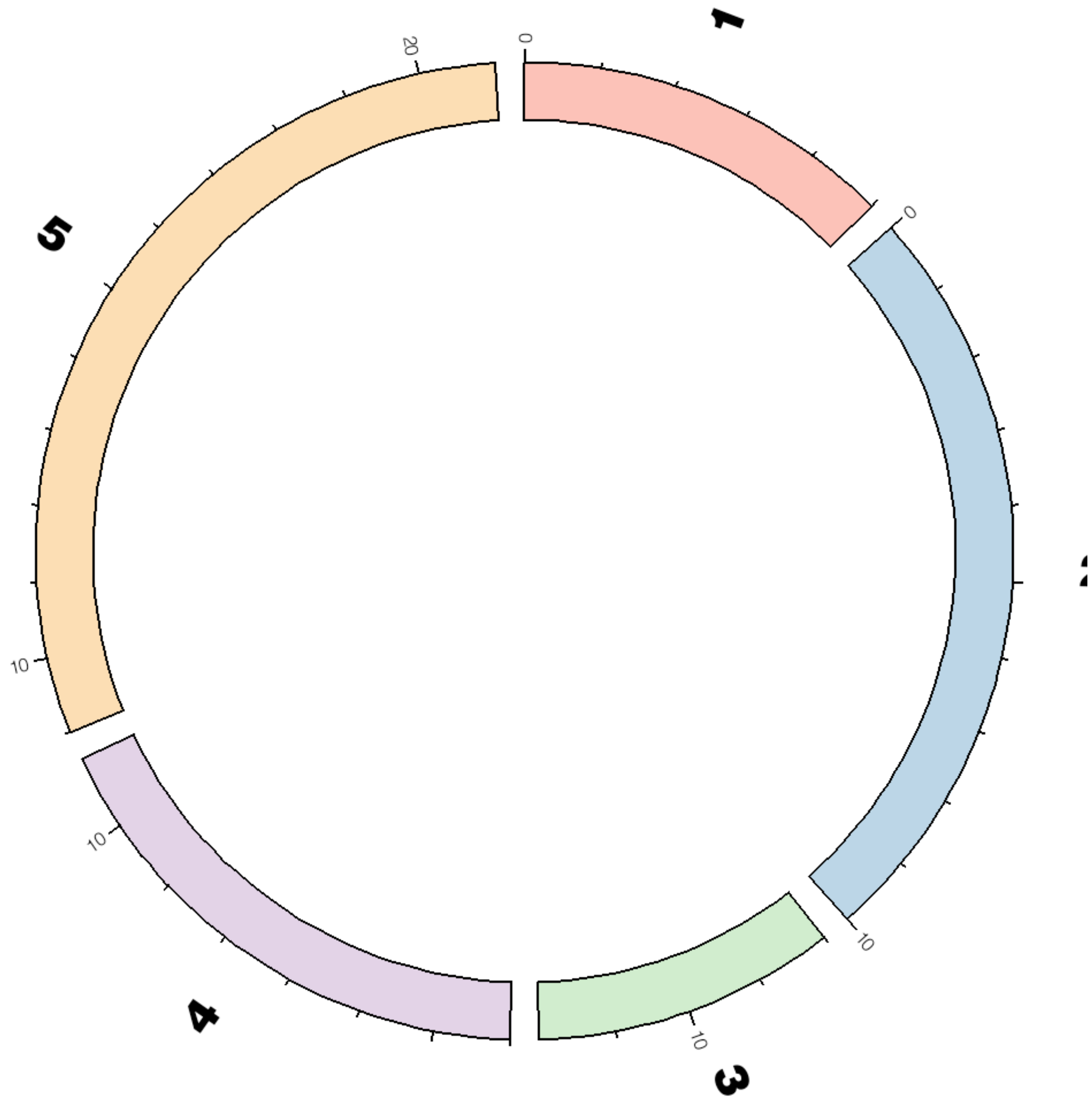


FIGURE 18

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
```

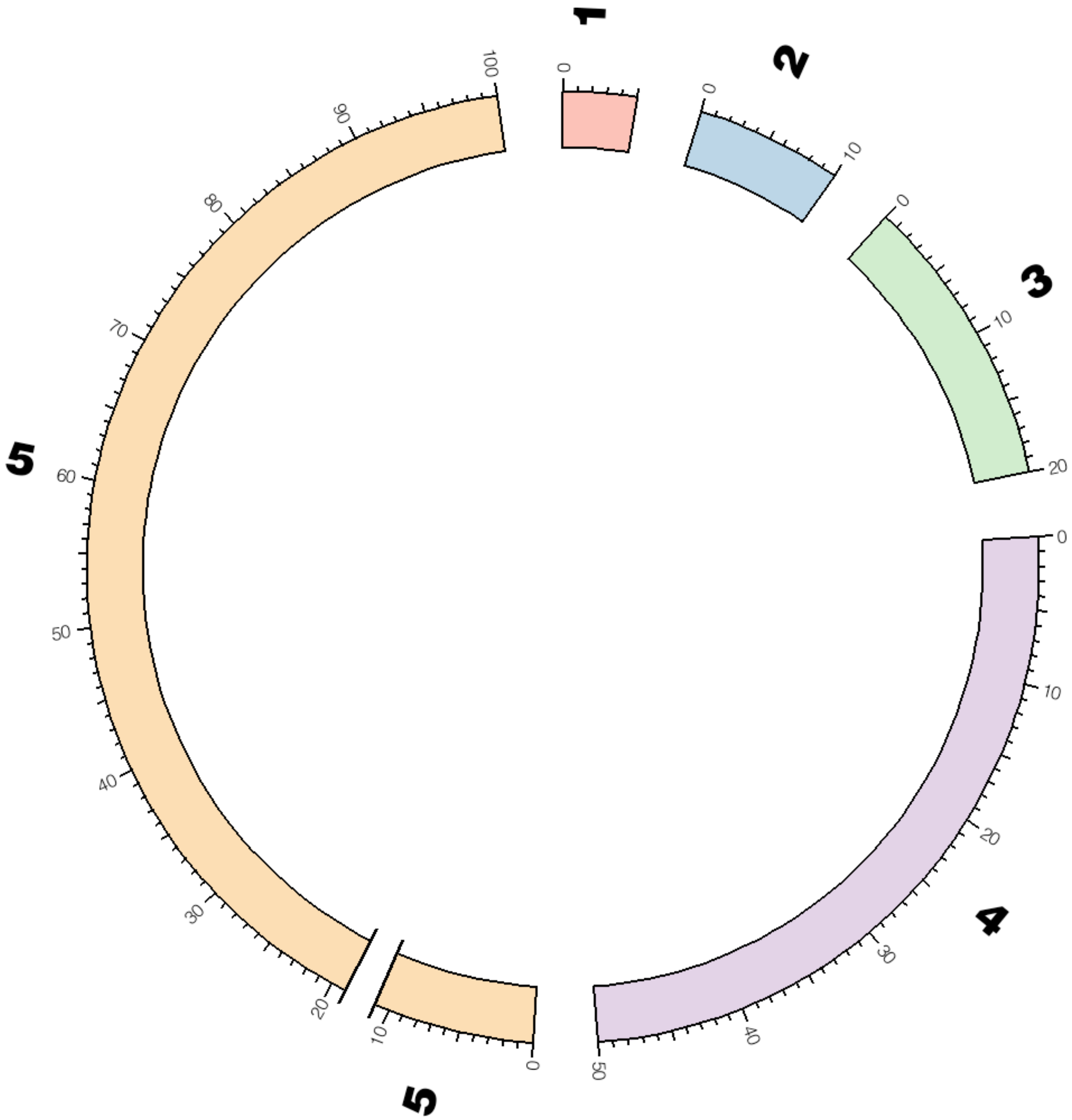


FIGURE 19

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.

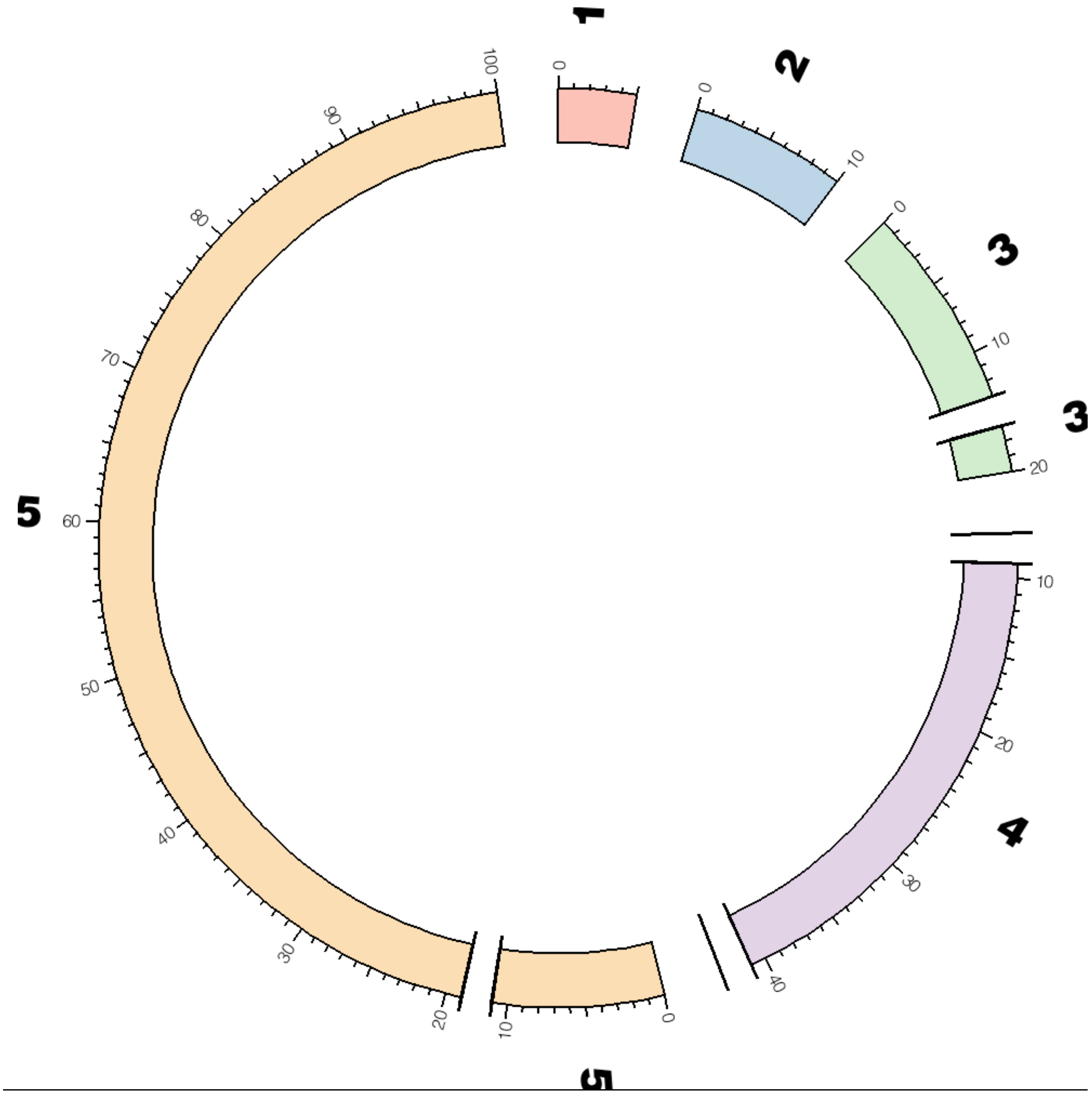


FIGURE 20

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.

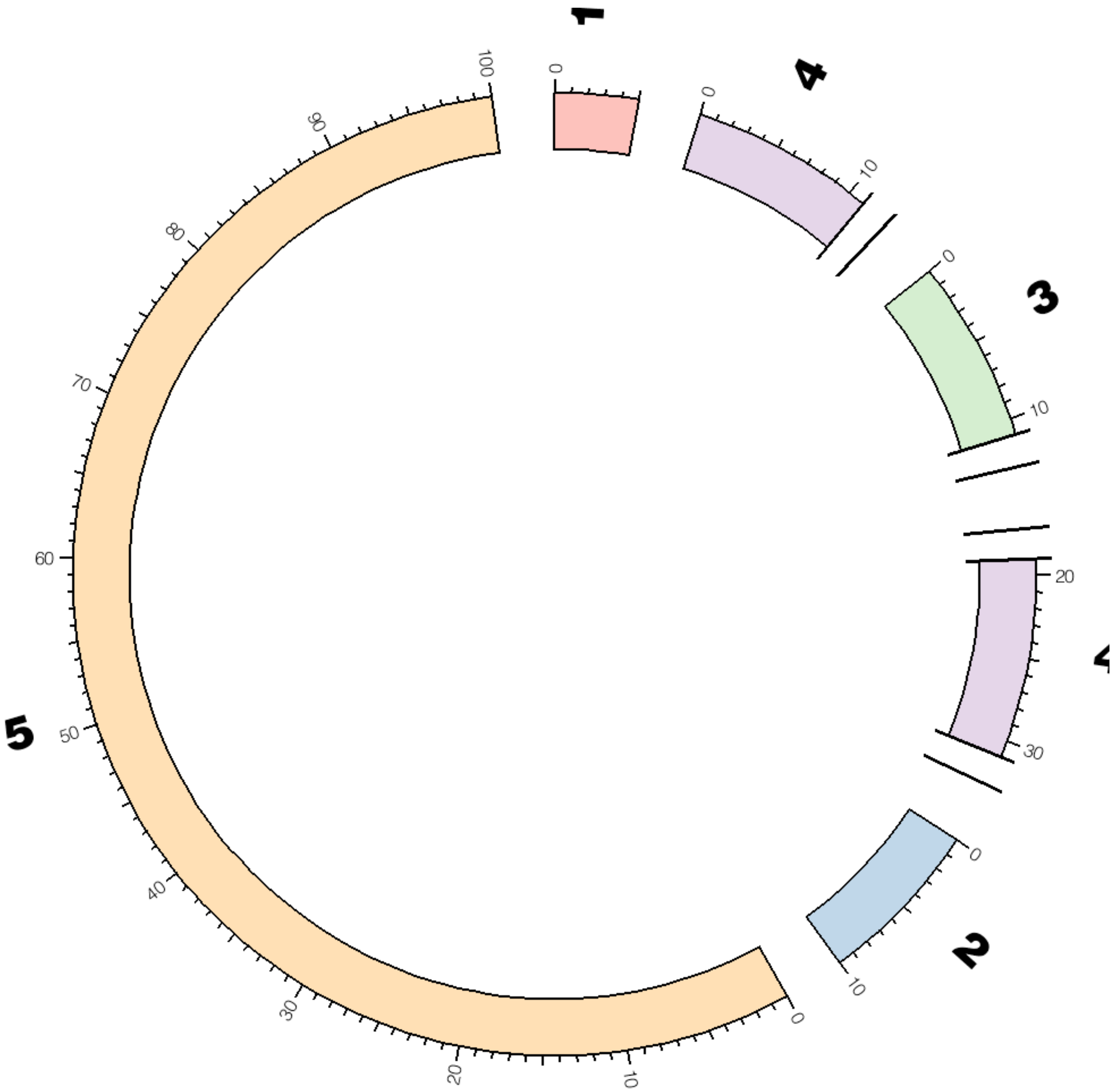
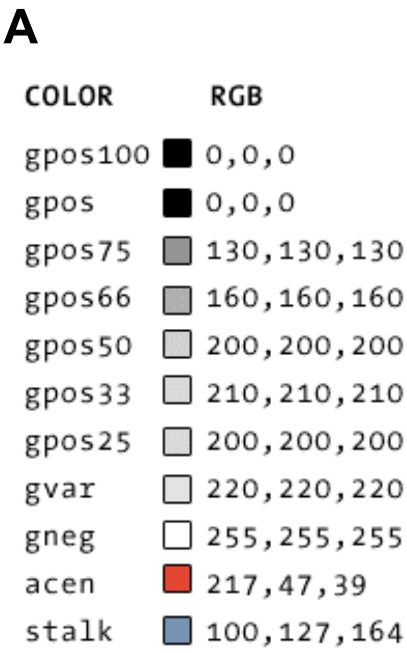


FIGURE 21

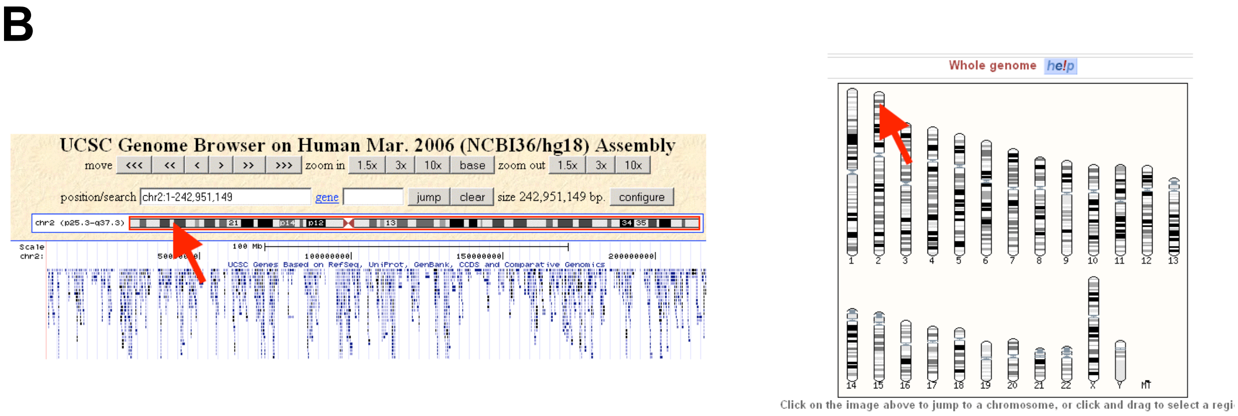
A chromosome can be split into multiple, independently ordered ideograms by using tags in the `chromosomes` parameter to associate a unique name to each ideogram.

```
chromosomes = chr3:0-11;chr4[a]:0-11;chr4[b]:19-31
```

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



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
UCSC Genome Browser on Human Mar. 2006 (NCBI36/hg18) Assembly
move
zoom in
zoom out
position/search chr2:1-242,951,149
gene
jump
clear
size 242,951,149 bp.
configure
Whole genome
help
Click on the image above to jump to a chromosome, or click and drag to select a regi

human chr2 at UCSC Genome Browser
http://genome.ucsc.edu

karyotype entry point to human genome at Ensembl
http://www.ensembl.org/Homo_sapiens/Location/Genome

FIGURE 22

(A) Band colors are defined in `color.conf` 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).

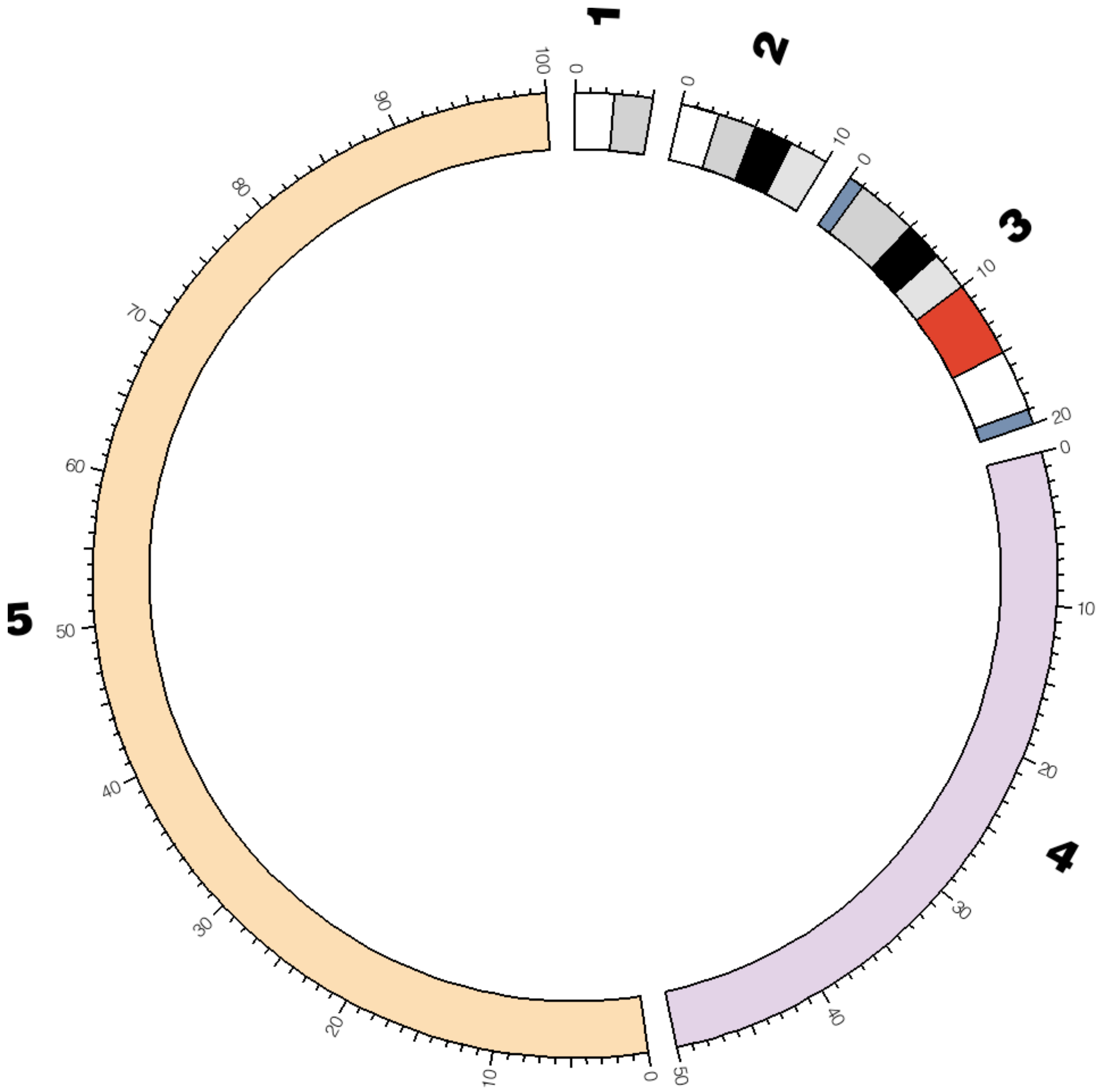


FIGURE 23

Cytogenetic bands appear as stripes within the ideogram segment.

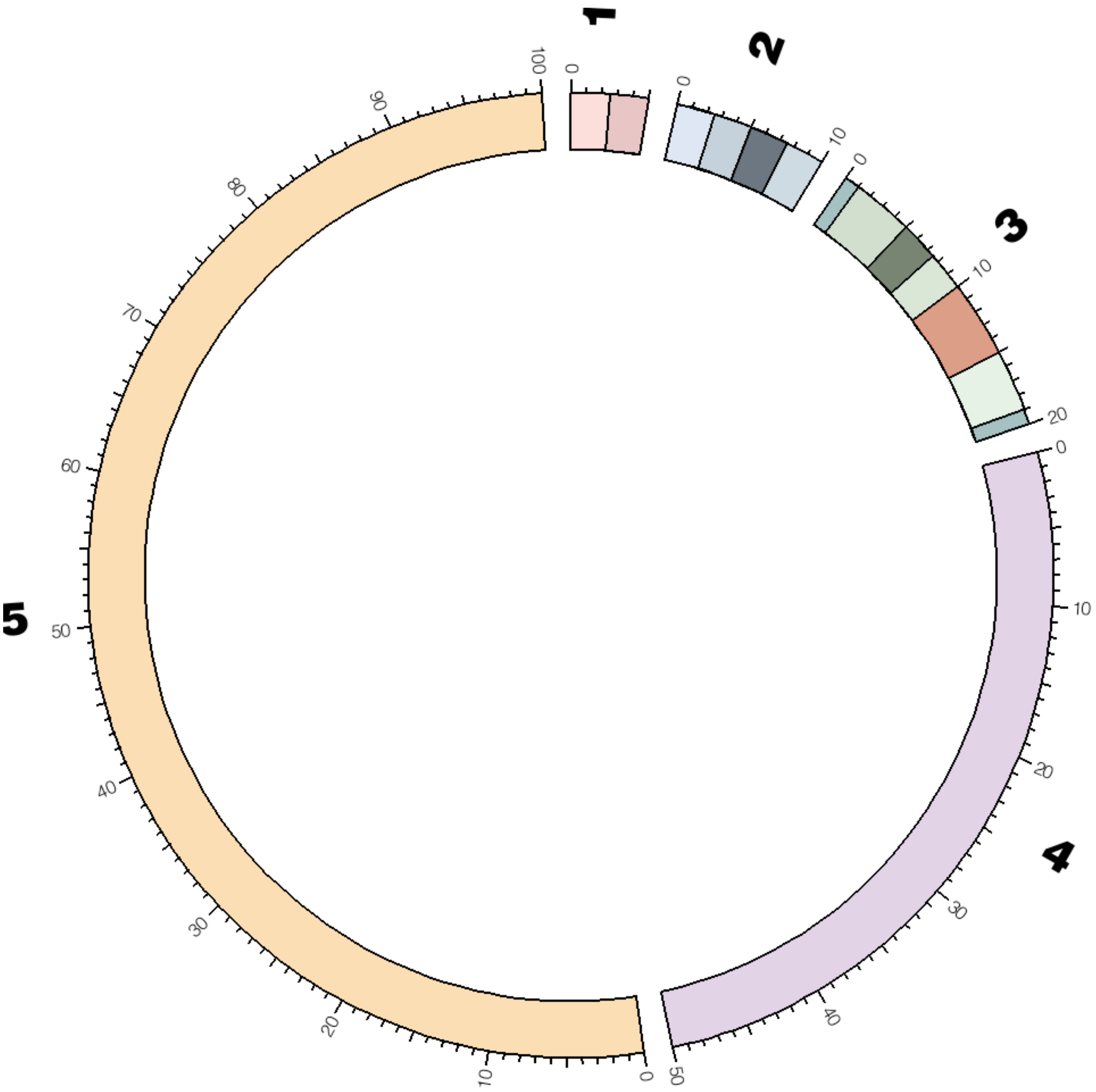


FIGURE 24

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`.

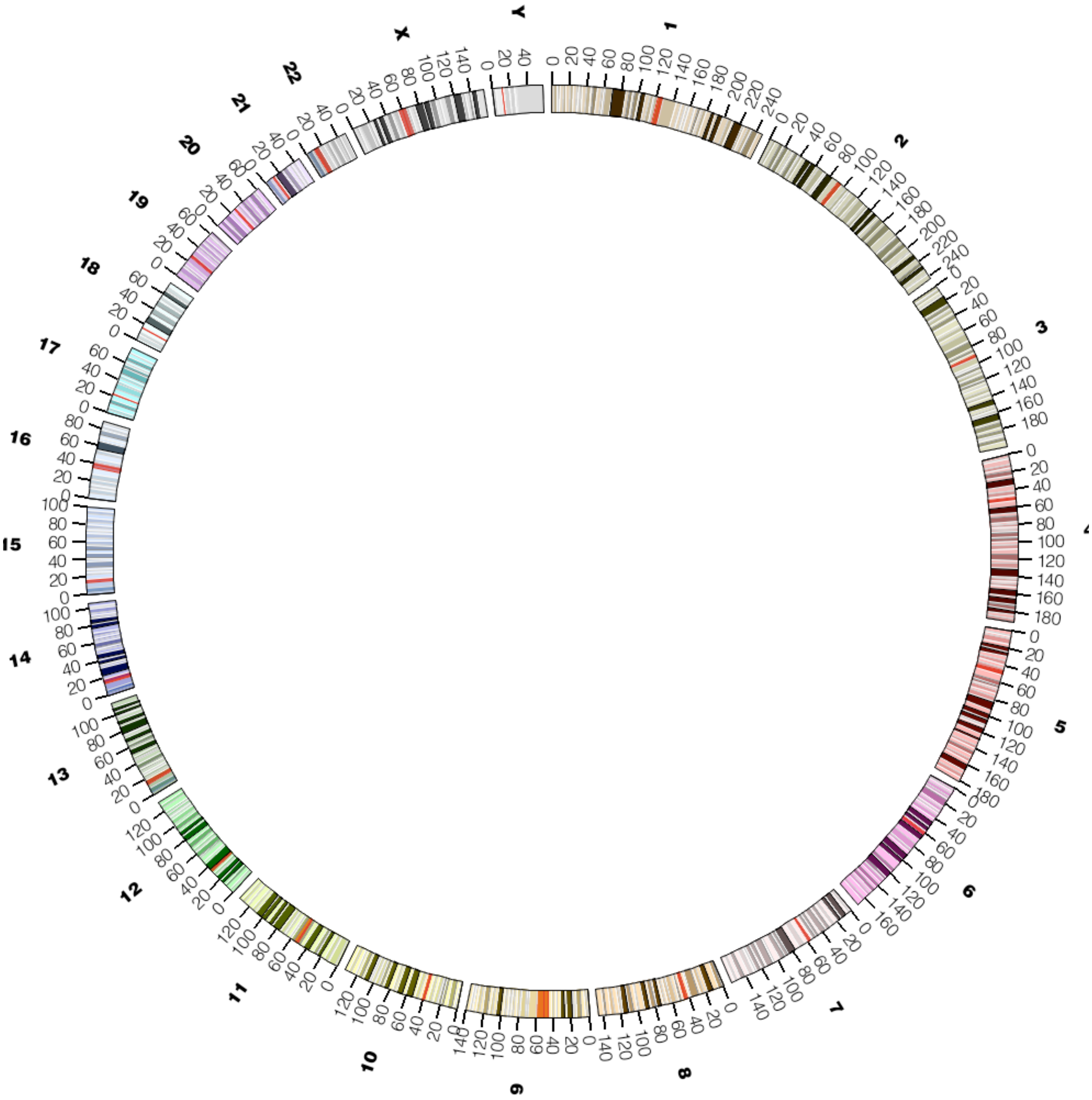


FIGURE 25

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).

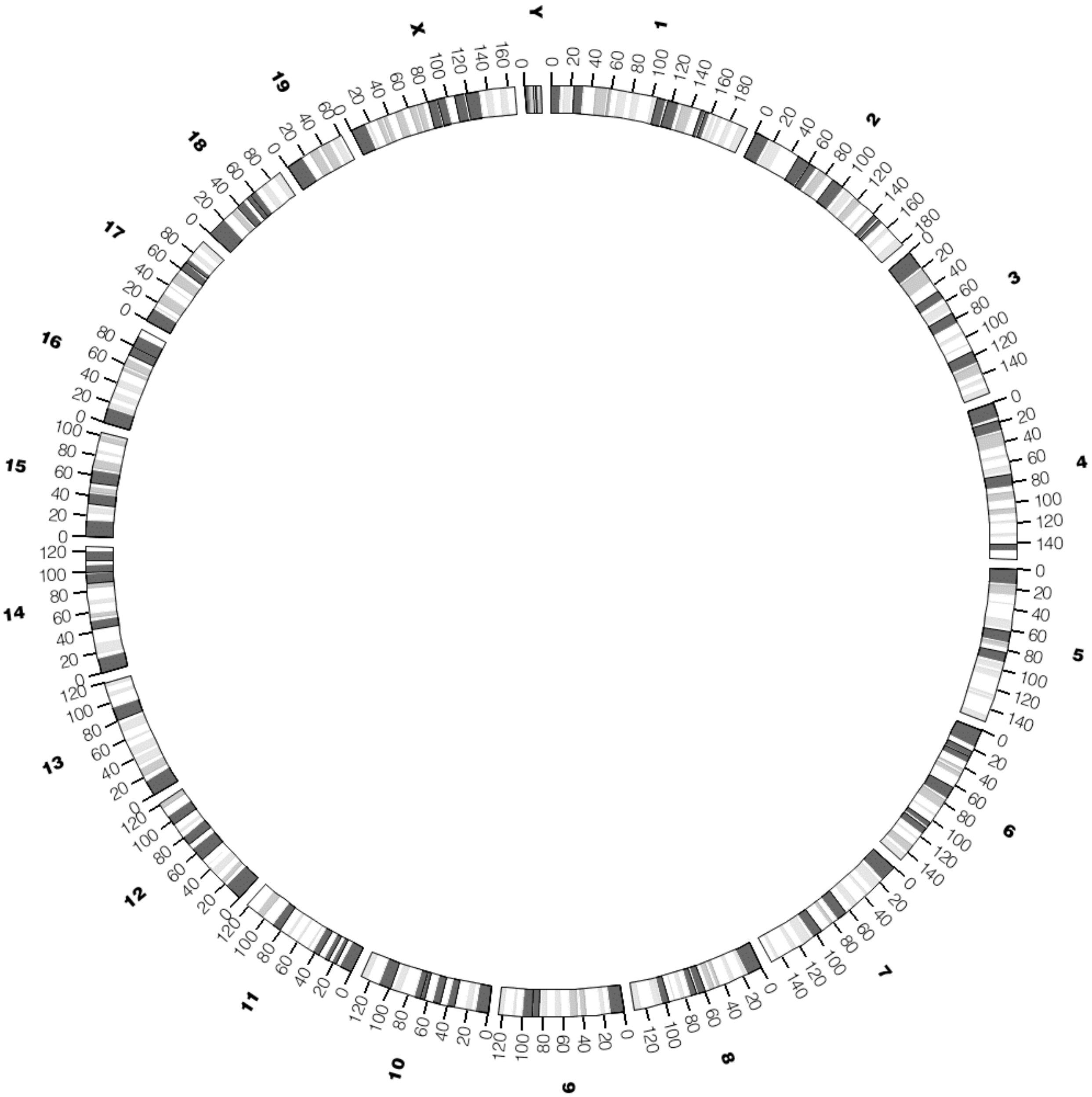


FIGURE 26

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.

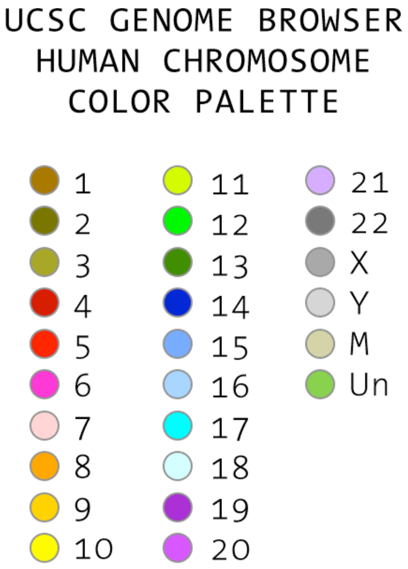


FIGURE 27

Conventional color palette for the human genome used by the UCSC genome browser. 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).

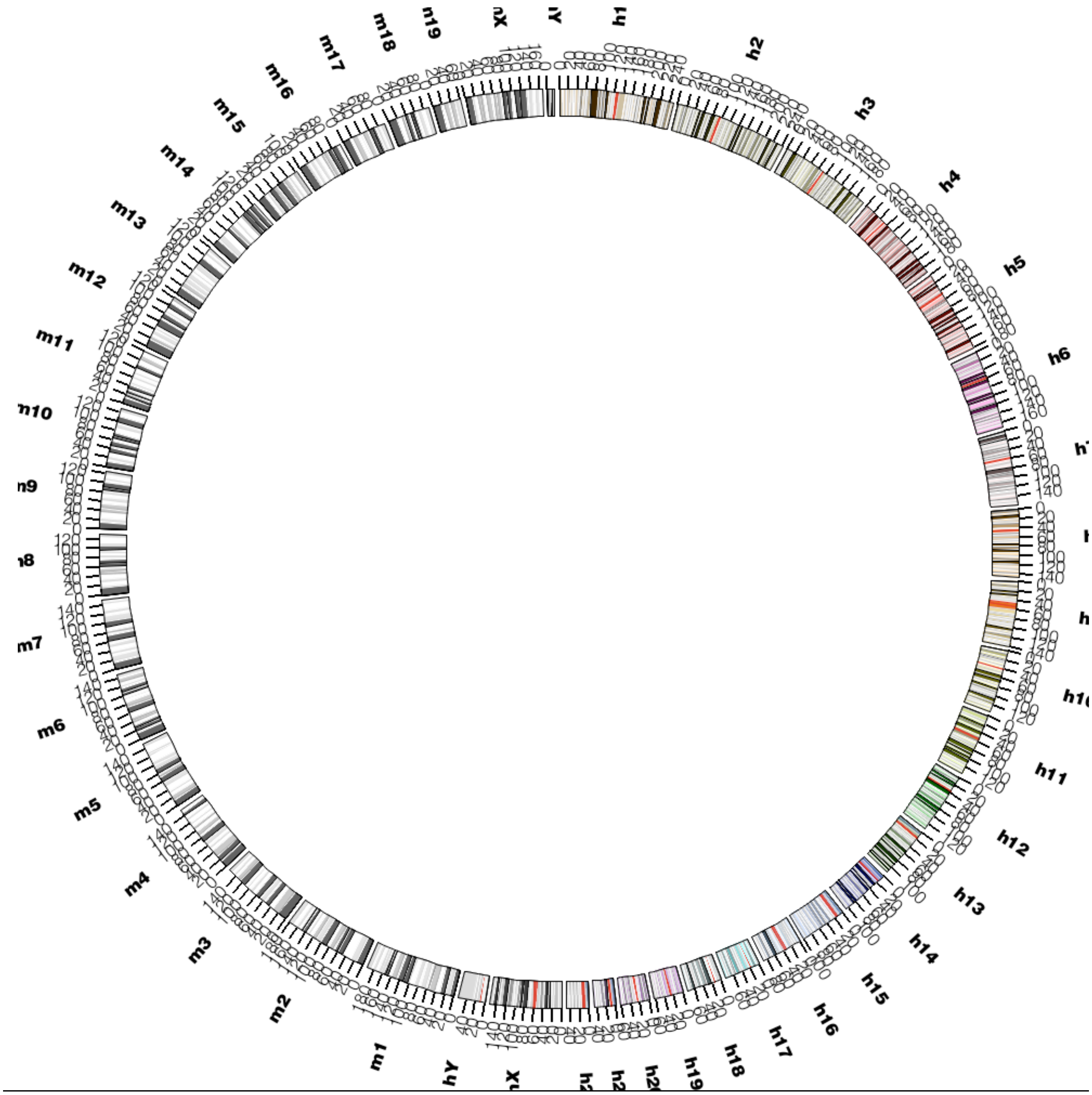


FIGURE 28

Ideograms of all human (labeled `hN`) and mouse (labeled `mN`) chromosomes.

Human ideograms are shaded by the UCSC genome browser color convention.

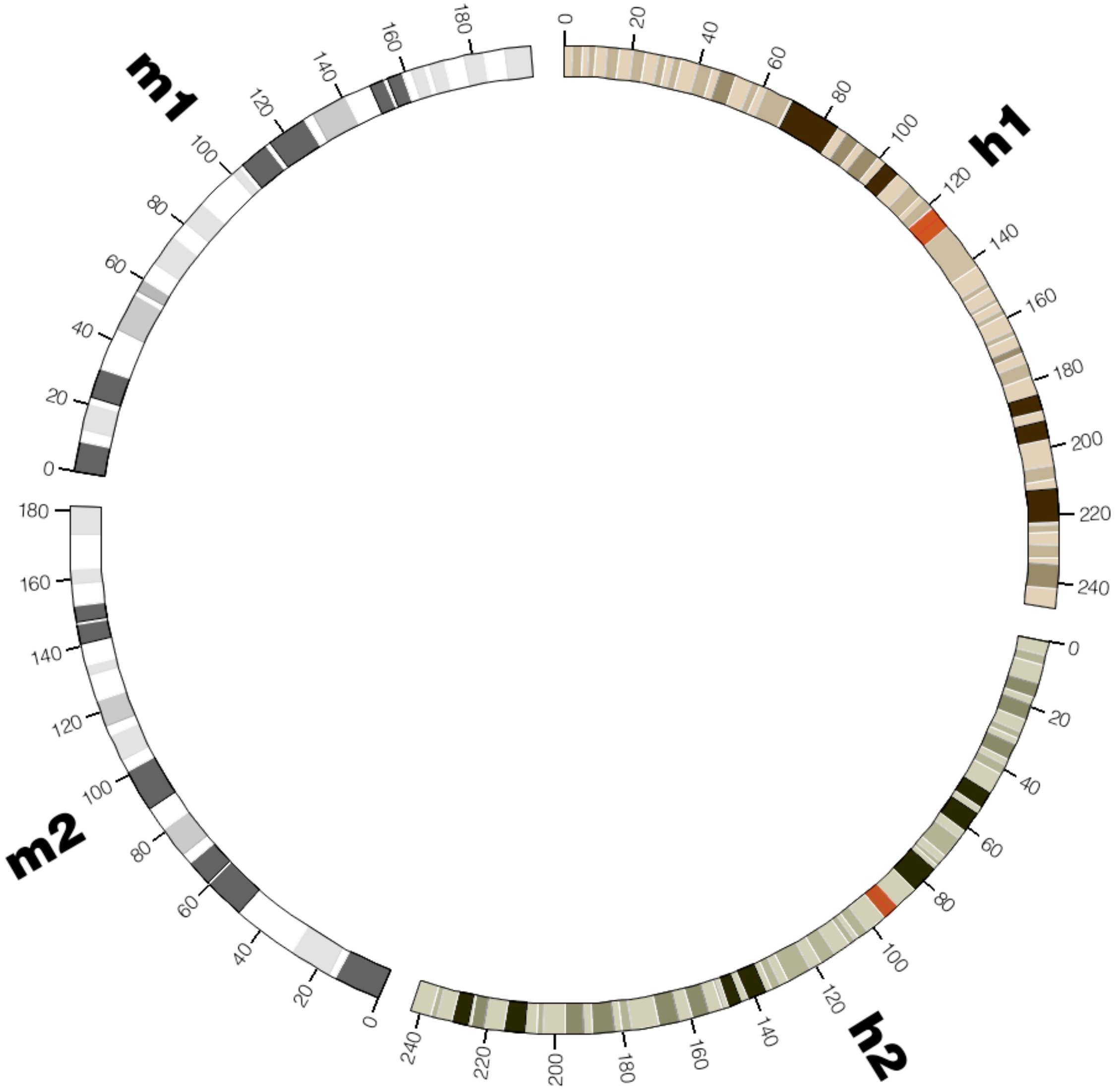


FIGURE 29

Chromosomes 1,2 of human and mouse genomes.

The ideograms start the top of the figure and progress clockwise.

The scale of each ideogram is oriented clockwise.

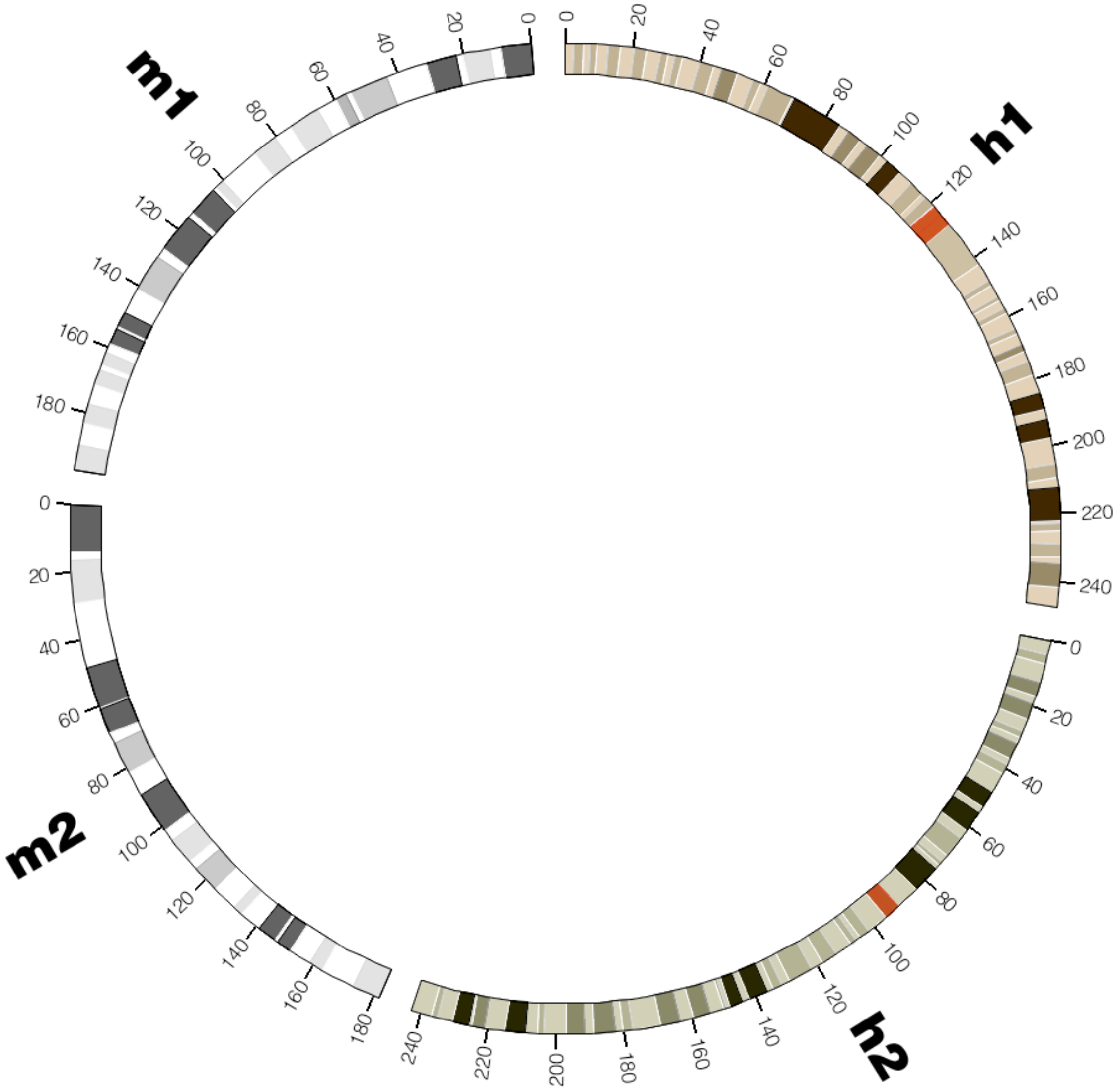


FIGURE 30

Chromosomes 1,2 of human and mouse genomes.

Scale on mouse ideograms is oriented counter-clockwise, opposite to the scale on the human ideograms. This makes adjacent the start and end of chromosomes 1 and 2 from the two species.

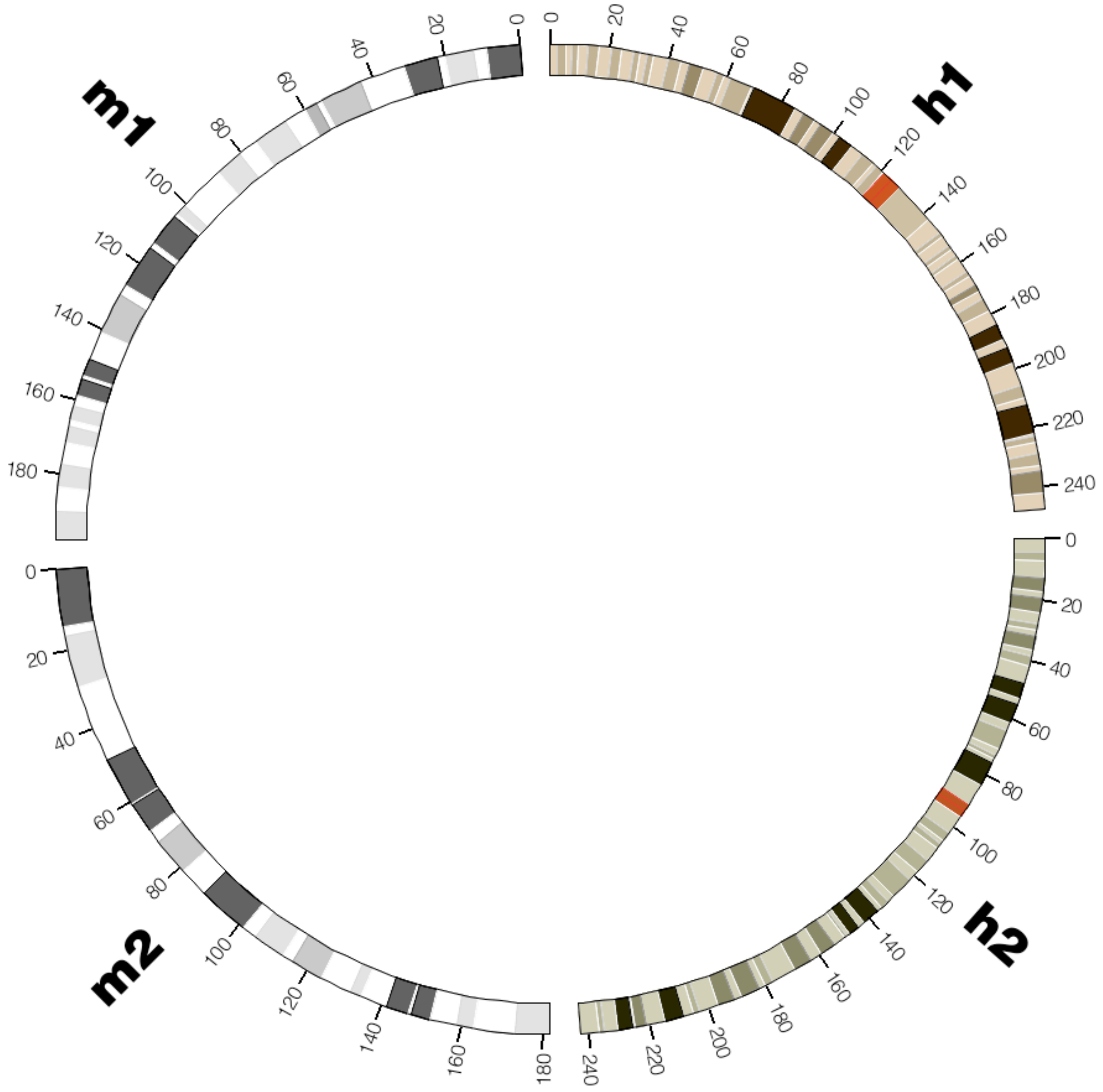


FIGURE 31

Chromosomes 1,2 of human and mouse genomes.

Scale of each ideogram has been adjusted so that it occupies ¼ of the figure.

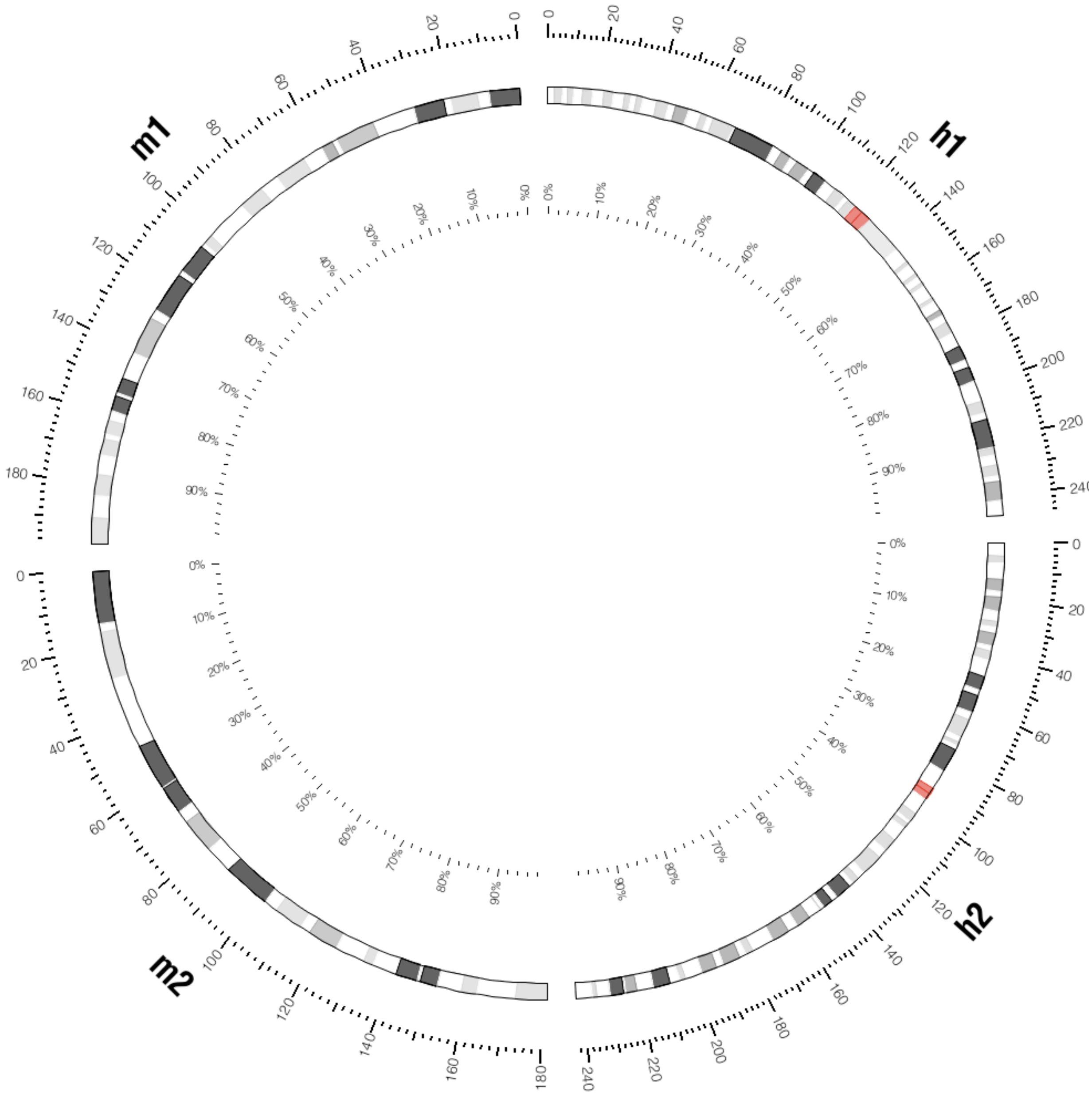


FIGURE 32

Chromosomes 1,2 of human and mouse genomes, with adjusted format parameters to make the ideograms thinner, shrink the ideogram circle and remove ideogram color.

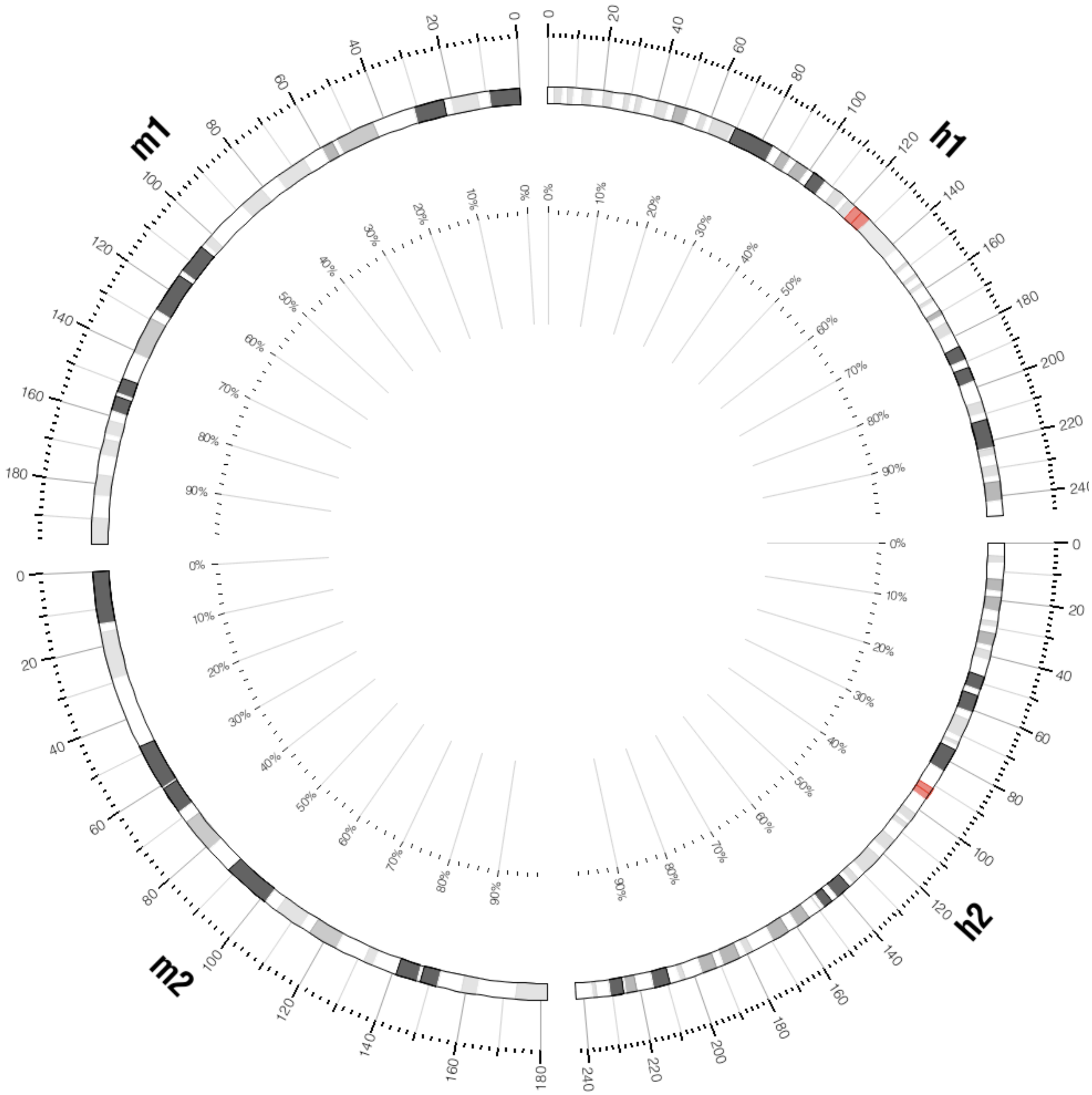


FIGURE 33

Chromosomes 1,2 of human and mouse genomes, as shown in Figure 32, but with an additional set of grids.

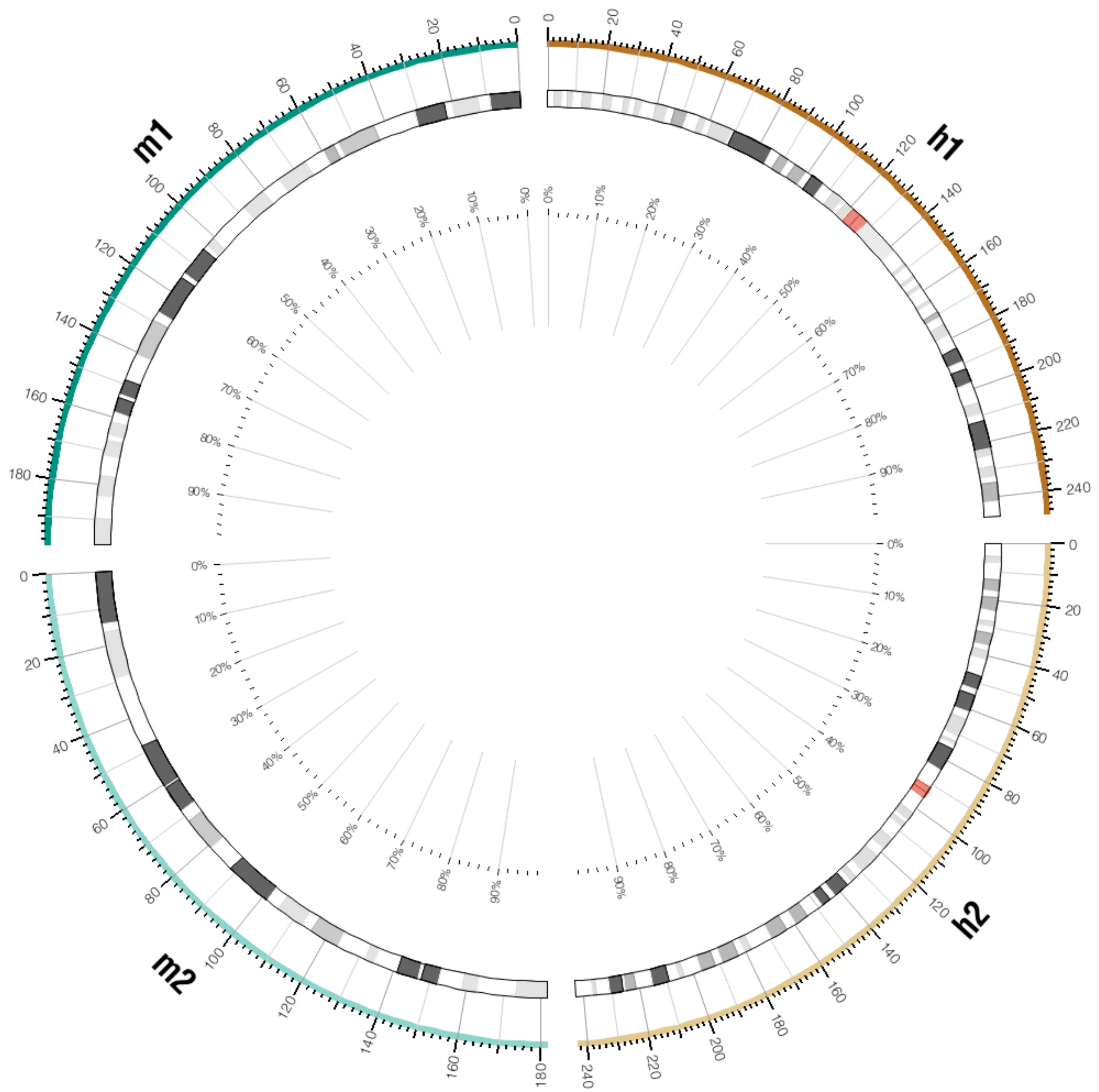


FIGURE 34

Chromosomes 1,2 of human and mouse genomes, as shown in Figure 33, but with highlight regions placed immediately inside the outer tick ring.

The regions provide a visual color index for the ideograms.