Version History
v0.15 1 Jul 2010
v0.14 30 Jun 2010
v0.13 30 Jun 2010
v0.12 29 Jun 2010
v0.11 29 Jun 2010
v0.10 27 Jun 2010

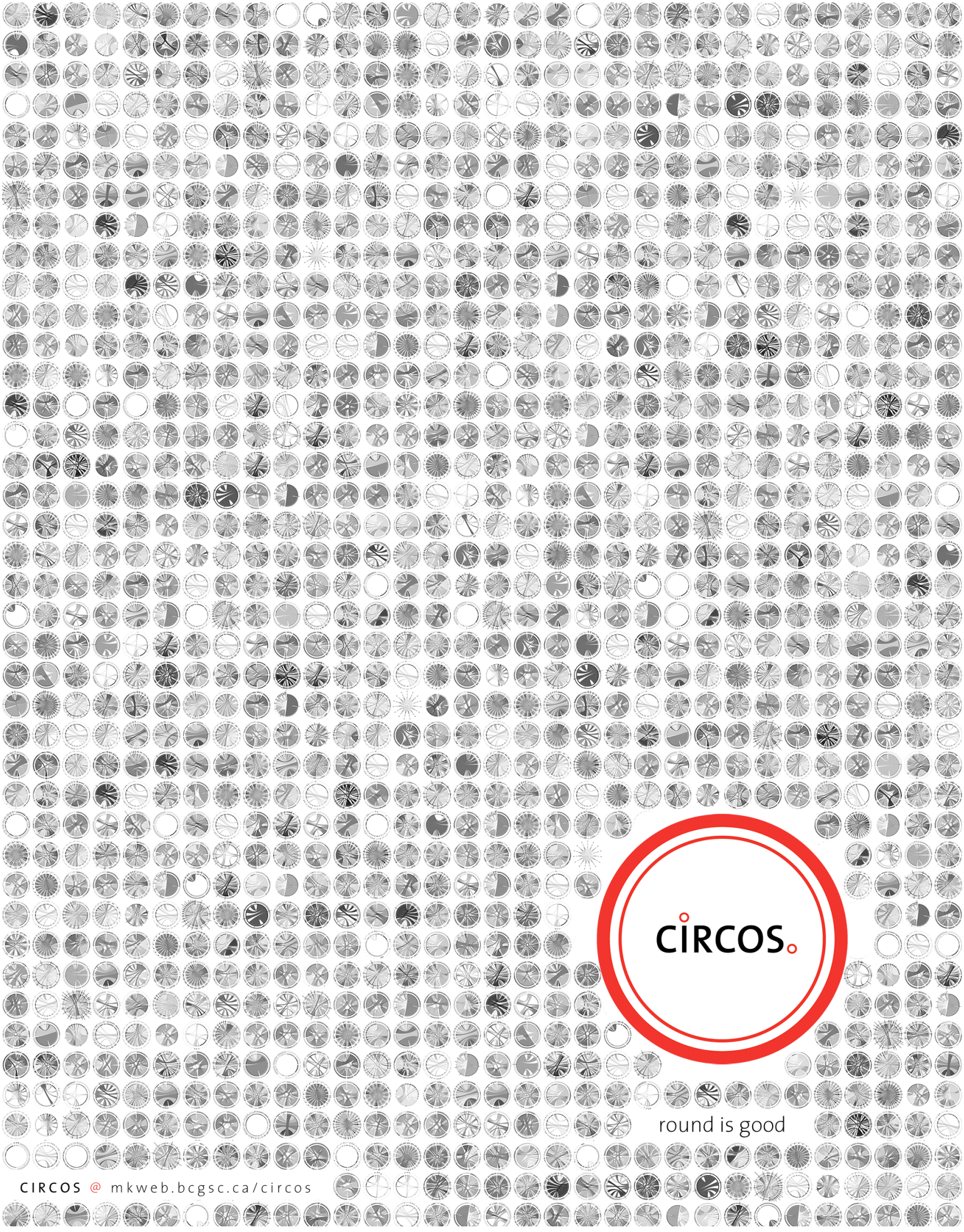

CIRCOS.
round is good
CIRCOS @ mkweb.bcgsc.ca/circos

Table of Contents

About this Introduction

In this document, you will be introduced to Circos by way of a tour of its capabilities. Using both synthetic and published examples, you will learn how parts of a Circos figure such as ideograms, data tracks, and highlights can be combined to effectively communicate complex information.

In the first section of this introduction, you will learn about the kinds of visualizations are possible with Circos. Many of the features that I will be drawing attention to will be covered in subsequent practical sessions, which will focus on the details of how to prepare input data files, write configuration files and generate images.

In the second section of this document, I'll provide a short description of Circos' architecture and configuration syntax. Because the practical sessions delve into the details of the configuration files, the introduction here will be high-level. With the knowledge you will gain from the practical sessions, and a little experimenting, you will be able to understand the configuration files for all the images in this document.

Finally, I will cover some principles of information design and apply them to figures from the literature. For many, I have created a redesigned version of the figure to explicitly demonstrate how to address common challenges such as representation, excess ink and color selection.

Creating Images

All Circos images in this document can be created using the configuration files found in `sessions/1`. The figure caption indicates which directory corresponds to the image (e.g. `sessions/1/5`). To create the image,

```
> cd sessions/1/5
> circos -conf etc/circos.conf
...
created image at ./s5-5.png
# if you've setup the local environment, an alias was defined
> runcircos
...
created image at ./s5-5.png
```

In some cases there are several configuration files (e.g. `circos.image-01.conf`), in which case you would use those

```
> circos -conf etc/circos.image-01.conf
```

For more information about the file structure of the lesson files, see the Session 2 preamble (`circos-s2-preamble.pdf`).

What is Circos?

Circos addresses the following problem*:* How does one show comparative information (that relates genomic positions) for two or more genomes? A linear layout is inadequate. When data is composited circularly, links reveal patterns in relationships between genomic positions. Links can be used to signify sequence similarity or difference, or structural variation between genomes.

Circos generates circular visualizations, of genomic and other data. It has been designed for creating publication-quality figures for presentation and print. The circular form is easier to visually follow (Figure 1), can accommodate data shown at a range of spatial resolution (Figure 2) and sits more compactly on the page. Circos implements various features that help in communicating large genomic data sets, including global and local scale stretching, axis breaks and formatting rules (Figure 3).

Circos can adjust the visualization based on data values. Data tracks can be associated with rules, which are snippets of code which act to change how data is displayed based on its characteristics (value, position, color, etc). These rules make it possible to highlight, alter or hide information without having to define this formatting in the data files.

Circos can be easily automated and incorporated into data analysis pipelines. The configuration file can be generated by another script, or through a web form, and the output image can be incorporated into a report, web page, or embedded in another image.

Circos does not have an interface. It is controlled entirely through a plain-text configuration file. The file defines image settings, location and format of data sets, and other formatting parameters.

Circos does not perform any analysis. It is a pure visualization tool. Circos can display a wide range of 2D data types (scatter plots, histograms, heat maps, tiles, etc), as well as links which connect two genomic positions. However, you are responsible for parsing your primary data, carrying out appropriate analyses and formatting it into a format that Circos understands.

Chromosomes and Ideograms

In Circos images, *ideograms* refer to graphical representations of *chromosomes* (or regions of chromosomes). By default ideograms are arranged in a circle (Figure 4), but radial positions of individual ideograms can be changed (Figure 4F).

The legibility of the figure depends on an ideogram organization that suits the data type and density. Several features in Circos help you create an ideogram layout that complements your data and.

Ideograms are drawn based on a karyotype file which defines the name, size and label of all chromosomes in the data set. Color, spacing, thickness, cytogenetic banding and radial position of ideograms can be independently adjusted (Figure 4).

Comparisons of multiple ideograms (possibly of chromosomes from various genomes) can be made more effective by rearranging ideogram order, orientation and scale. This is shown in Figure 5, where these properties of four ideograms are adjusted.

The ability to crop and rearrange ideograms is very helpful when only a small part of the genome confined to short regions is being shown (Figure 6).

Scale and zooming

You have already seen in Figure 3 and Figure 5 how the scale of an entire ideogram can be changed to magnify (or shrink) the ideogram. This feature is further extended by allowing you to limit the scale change to an ideogram region. This is shown in Figure 7, where chromosomes 1 and 2 are shown with their regions shown at different magnification.

The ideogram of chromosome 1 is divided into regions at *1x*, *2x*, *3x*, *4x* and *5x*. For chromosome 2, regions with a decrease in scale are created. Note that the scale change in this figure is applied within an ideogram – you do not need to split the chromosome into individual

ideograms to locally change scale (although you could, by means of changing the global scale of a cropped ideogram).

Zoom regions that change magnification can overlap, in which case the largest scale change (from *1x*) is used.

In Figure 8, 13 zoom regions are defined (6 on chromosome 1 and 7 on chromosome 2). This scale stepping can be automated by defining a region that has a continuous scale adjustment. To do this, you specify the region you wish to adjust and its new scale, and then specify how Circos should vary the scale in the neighbourhood to make the scale transition smoother.

An example of smooth scale transition is shown in Figure 8, where two regions have a change in scale (chr1:120-125 Mb *10x*, chr2:120-125 Mb *0.1x*). No other scaled regions are explicitly defined, but Circos uses scale smoothing to automatically adjust the magnification in the neighbourhood of these regions. This feature is very useful if you wish to draw attention to a region, and show its data in higher detail, but do not want a hard scale boundary.

Global and local scale transformation can be combined. This is shown in Figure 9. Chromosome 2 has been split into three ideograms, with a magnification of *2x*, *2x* and *0.5x*, respectively. Chromosomes 1 and 3 are shown at *1x*. Within each ideogram, independent regions of scale change are defined. The histogram track inside the figure shows the level of magnification for a region (each grid line represents change of magnification by *1x*). Local scale smoothing results in a stepping of magnification (chr1:120-125 Mb) or reduction (chr2:78-82 Mb). These local scale transformations are combined with the global scale change.

Data Tracks

Track Layout

Data tracks can be placed anywhere within the image, including on top of ideograms. Tracks are confined to a radial range and may overlap (Figure 10). The benefit of overlapping tracks will be described below.

Most of the data tracks shown in Figure 11 are confined to a radial range (annulus). For example, a histogram track might be placed within 50%-75% of the circle's interior. Link tracks (Figure 11 D,E,O,Q) are special in that they can fill a circle (if they are curved inwards) or occupy an annulus (if they point outwards). This flexibility is shown in Figure 17.

Track Type

Circos supports a large number of track types, shown in Figure 11. Many are interchangeable, meaning that they can use the same input file (e.g. scatter, line, histogram and heat map plots).

Some tracks, such as the text track, can be turned into a glyph track by using a font of symbols (Figure 11A).

By applying rules to dynamically alter how data points are displayed, together with overlapping tracks and transparency, large amounts of information can be effectively layered in the figure.

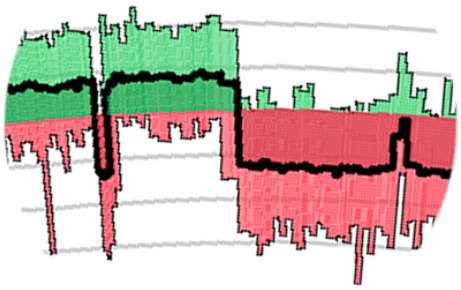
Overlapping Tracks

Compound tracks can be constructed by overlapping data tracks. For example, in Figure 12 the yellow/green histogram is constructed from three separate histograms, each with a different fill and outline color. The bottom-most histogram that is drawn (yellow) displays the maximum value of the data track within a position window. The next histogram that is drawn is the green

histogram, which displays the average value. This histogram has a white outline, which acts to separate the colors of the two histograms and provide a measure of the average. The last histogram, which stores the minimum value within a window, is not visible because it has a white fill. The purpose of the last histogram is to clip the green histogram to make it appear to down only as far as the minimum value.

Another application of overlapping tracks is shown in Figure 13, where the same data are shown at two resolutions. The first layer of tracks is formed by the green and red histograms which emanate from a common baseline.

The red histogram drops down and denotes the minimum copy number value within a window. The green histogram is oriented outward and shows the maximum copy number value. The thick black line, drawn on top of both histograms, represents the average copy number value.



Individual points in the copy number data set cannot be drawn at the resolution of the entire genome because the data set consists of 500,000+ values (Affymetrix Mapping 500K array). Instead, histograms are used to bin copy number values within a window and provide a measure of min/avg/max statistics.

However, when ideogram scale is increased, such as illustrated in Figure 5D, individual copy number values can be shown. This is demonstrated in Figure 13, where two zoomed regions of chromosome 17 contain both the histograms and individual copy number values, which are shown as a scatter plot on top of the histograms.

Partitioning Tracks

You can creatively color histograms by partitioning a track into multiple components. This approach is illustrated in Figure 14 and the result in Figure 15. Let's walk through how partitioning works and then look at the result.

The principle is based on decomposing a single track (Figure 14A, track A1) into three independent adjacent tracks (Figure 14B, tracks B1, B2, B3). Tracks B1-B3 together occupy the same region of the figure as track A1. Moreover, they also span the same data range, although individually they represent only a portion of that range.

In this example, track A1 has a width of *0.3r*. Tracks B1-B3 partition the track into three, thus each has a width of 0.1r. Track A1's data range is `[-0.3,0.3]`, which is partitioned into [-0.3,-0.1], [-0.1,0.1], [0.1,0.3] in tracks B1-B3.

The same data file is used to populate tracks B1-B3. Histogram values that fall outside of the data range of a track result in bins that are clipped at the track's top and/or baseline. For example, a histogram bin of size `0.25` would start in the middle of track B2 and reach its outer edge (clipped at 0.1). In track B3 the bin would start at the baseline (clipped at 0.1) and terminate within the area of the track (since bin value is less than track maximum range, 0.25 < 0.30).

If each of the tracks B1-B3 is given a different background and data color values, the resulting histogram will be composed of bins that are composed of parts with one or two colors. This is shown in Figure 15. This effect cannot be achieved by rules, because although rules can adjust the color of a bin based on its value, the color of the *entire* bin is affected.

With this partitioning scheme, a bin is split into multiple parts (by partitioning it into multiple tracks). The same result can be achieved using a stacked histogram, but this approach would require that the histogram data file be reformatted and, inconveniently, the value regions into which the data is split would be hard-coded in the data file.

Using Connectors to Match Multiple Scales

It is common in genomics for data set values to sample the genome non-uniformly (e.g. at gene positions). In such data sets, many values cluster in one location, surrounded by relatively large regions for which no data is present. Drawing these kind of spatially non-uniform data sets can be challenging, because the data points do not effectively fill the space in the figure.

An example of this is Figure 16, which shows methylation information on human chromosome 22 for 7 different tissues. Methylation is determined only at specific positions, identified by small orange tiles (Figure 16A). In order to make full use of the figure area, methylation values are remapped onto a uniform scale (using the index of the position) and binned into histograms (Figure 16C). The spatial relationship between the methylation probes and the remapped scale is shown using a connector data track (Figure 16B).

The example in Figure 16 can be expanded to include another scale, matched by a connector track to the existing physical scale (methylation probe positions) or the rescaled content (binned methylation values).

Link Geometry, Rules and Bundling

Links are Circos' selling feature. Drawn as quadratic Bezier curves, links create a visual association between two positions and are effective at showing synteny, alignments, rearrangements (translocations, fusions, or any event that brings distant regions into adjacency), and any kind of similarity (or difference) relationship.

Figure 17 shows six ways to visualize the same link data set – segmental duplications within the human genome. It is typical to have a large number of links to cope with and Figure 17 demonstrates some approaches to mitigate the complexity of numerous links.

Rules can be used to change the visibility, color and thickness of links, such as in Figure 17B, where only inter-chromosomal links are shown, with line thickness and darkness proportional to the size of the segmental duplication. Figure 17C applies rules which only show (a) intra-chromosomal links, whose geometry is adjusted to place them outside of the ideogram circle and (b) links that involve chromosome Y. For both groups, links are colored by chromosome. Note how link geometry in Figure 17C differs from Figure 17B – in Figure 17C links contact the ideogram perpendicularly, with a defined bulge in the middle. Figure 17D uses a few more rules to include links that involve chromosome 8, with this group drawn faintly in the background.

To make interpretation of a large number of links easier, Circos has a tool to combine adjacent links into larger links into bundles. Bundles are still links, in the sense of how they are stored and shown, but their size and position is calculated from the original link file. Thus *link* and *bundle* can be used to distinguish between the original link data and the processed link data. Figure 18 illustrates the principle of bundling links.

An example of bundling is shown in Figure 17E and Figure 17F, which show the bundles formed from links in Figure 17A. In Figure 17E, bundles are drawn in transparent black and in Figure 17F they are colored by chromosome color (for this example, it is arbitrary whether the chromosome at the start or end of the link is chosen).

Two published examples of combining rules and bundles are shown in Figure 19, which displays synteny between a subset of human and dog genomes and Figure 20, where several layers of bundles are stacked together to give the figure a more artistic feel.

Image Panels

The flexibility of a plain-text configuration file and rules allows for automation. By invoking Circos multiple times, once for each image, and using an image compositing tool (such as Image Magick's command-line `convert` utility), a panel of images can be created.

Two such examples are shown in Figure 21, where individual images represent the comparison between one dog (or human) chromosome and the entire human (or dog) genome.

Track Input

Data for tracks is loaded from a plain-text file. Each data point is stored on a separate line, except for links which use two lines per link.

The definition of a data point within a track is based on the genomic range, which is a combination of chromosome and start/end position. For example,

```
# the basis for a data point is a range
chr12 1000 5000
```

All data values, regardless of track type, will be positioned using a range rather than a single position. To explicitly specify a single position, use a range with equal start and end positions.

Tracks such as scatter plot, line plot, histogram or heat map, associate a value with each range. The input to this kind of track would be

```
# scatter, line, histogram and heat maps require a value
chr12 1000 5000 0.25
```

The exception is a stacked histogram, which associates a list of values with a range.

```
# stacked histograms take a list of values
chr12 1000 5000 0.25,0.35,0.60
```

The value for a text track is interpreted as a text label (other tracks require that this field be a floating point number).

```
# value for text tracks is interpreted as text
chr12 1000 5000 geneA
```

The tile track does not take a value – only a range.

```
chr12 1000 5000
```

Finally, links are a special track type which associates two ranges together. This track type requires that each range is specified on a separate line, with the pair identified by a common but unique identifier.

```
id123 chr12 1000 5000
id123 chr15 5000 7000
```

In addition to the chromosome, range and (if applicable) value, each data point can be annotated with formatting parameters that control how the point is drawn. The parameters need to be compatible with the track type for which the file is destined. Thus, a scatter plot data point might have

```
chr12 1000 5000 0.25 glyph_size=10p,glyph=circle
```

whereas a histogram data point might include the option to fill the data value's bin

```
chr12 1000 5000 0.25 fill_color=orange
```

Other features, such as URLs, can be associated with any data point. For URLs the parameter can contain parsable fields (e.g. `[start]`) which are populated automatically with the data point's associated property.

```
# the URL for this point would be
# http://domain.com/script?start=1000&end=5000&chr=chr12
chr12 1000 5000 0.25
url=http://domain.com/script?start=[start]&end=[end]&chr=[chr]
```

Tick Marks and Tick Labels

Circos has an extremely flexible tick mark mechanism, which allows you to place tick marks (and tick labels) anywhere in the figure. Well-placed and clearly labeled tick marks help the reader navigate around the figure. Ticks should be subtle, but clearly legible – they are a navigation aid and should not compete for attention with data in the figure.

Tick marks can be placed based on absolute position (every 5 Mb) or relative (every 5%). Tick labels can be freely formatted to be absolute (5 Mb), relative (5%), fixed precision (5.0 Mb, 5.00 Mb, 5.000 Mb) and have a suffix (e.g. 5/10, 5%, 5 Mb).

As shown in Figure 22, ticks are broadly divided into groups. Each tick group defines ticks at a fixed spacing (either absolute or relative). Ticks are drawn in order of decreasing spacing. Thus, for ticks in Figure 22 the 10 Mb ticks are drawn first, then 5 Mb ticks and then 1 Mb ticks (ticks that fall on the same position from different groups are drawn only once, from the group with the largest spacing, e.g. once 50 Mb tick is drawn from the 10 Mb spacing group, the 5 Mb and 1 Mb spacing group does not contribute to a tick here).

Ticks within each group are independent and can be formatted to have independent position, size, color, labels, etc.

The display of ticks from a group can be suppressed for chromosomes, or regions. This is helpful if you wish to reduce the complexity in part of a figure, where scale navigation is not important. This kind of filtering is shown in Figure 23, where ticks are suppressed on hs1, and within 0-100Mb on hs2 and 100-) on hs3.

In a figure where the scale of ideograms is not constant (e.g. some ideograms are magnified, while others are reduced), the density of ticks from a given tick group can vary. The result is that ticks can become very dense and overlap each other. To mitigate this, you can definite a minimum separation for ticks and, independently, for their labels. This separation ensures that if ticks from a group should be too close, they are not drawn. This dynamic tick and label suppression is automatic and is shown in Figure 24.

In addition to a physical scale in base pairs (5 Mb, 10 Mb, etc), some displays benefit from a relative scale, where it is more important to be able to identify fractional positions, such as 10%, 20% and so on. This kind of division is achieved using relative ticks, which can be spaced by a relative distance to the chromosome (e.g. every 1%). The tick label can be either relative (e.g. 1%) or absolute (e.g. 2.47, which is 1% of human chromosome 1). Absolute and relative ticks are shown in Figure 25.

Relative ticks can be further customized by changing their scale divisor from chromosome to ideogram. When a chromosome divisor is used, tick labels are determined based on their position within the chromosome, regardless where they fall within an ideogram. However, when an ideogram divisor is used, the tick is considered to be relative to the region shown by the ideogram. For example, if we have an ideogram that shows chr2:0-100 Mb, a relative tick at 50 Mb would be considered to be 50%, if the ideogram is used as a divisor. If the chromosome is used, then the tick would be 21% (50 Mb/243 Mb).

When many data tracks are drawn, it can be helpful to draw ticks at more than one position. Ticks from a tick group can be placed at any one or more radial positions. This is shown in Figure 26, where tick groups at 1, 2.5, 5, 10 and 50 Mb are drawn at 1, 2, 3, 4 and 5 radii each, respectively.

Because the linear scale within the figure naturally varies as a function of radial position, limiting ticks within the center of the figure to those that are spaced by a large distance maintains uniform tick density.

In addition to tick groups that place ticks at a specific spacing, a group can define one or more ticks at specific positions. This is useful for cases where you want to draw attention to scale positions using larger, or colored, ticks.

In Figure 27, there are four tick groups, each defining ticks are specific positions. You can assign positions based on relative or absolute values. The figure shows one group has ticks at relative positions of 0.05, 0.15, 0.20, but with labels having absolute positions (12, 37, 49 Mb). Another group has absolute positions of 30, 32, 34 and 40 Mb, with absolute labels.

Text

One of the most advanced tracks in Circos is the text track. This track associates a string with a genomic span and is useful for identifying text-annotated regions, such as genes.

```
hs1 47674275 47678950 FOXD2
hs1 63561317 63562754 FOXD3
hs1 47654330 47656310 FOXE3
hs1 42414797 42573490 FOXJ3
```

Labels are drawn oriented radially and may be associated with a line that connects the label to its position. Circos automatically arranges the labels, within an allowed range, so that they do not overlap. When the text track is dense, labels will stack to fill the radial extent of the track.

Figure 28 shows three text tracks of various density. Labels in each track have lines that point to the position of the label. Note that the label itself can be offset by any distance from the baseline of the track. For example, the baseline of the outer track is the top of the ideogram, but the labels are offset to allow room for ticks.

Figure 29 shows an example of a simple text track, which displays the name of the cytogenetic band. Labels that cannot fit are stacked, with a line connecting the label to its position.

When a symbol font is used for text tracks, this track type can be turned into a glyph track. To show how this works, Figure 30 shows the same text data (sequence) rendered in a text track using different fonts and rules.

```
...
hs1 139 139 G
hs1 140 140 T
hs1 141 141 C
...
```

In the first track, each sequence base (which is an individual label) is colored based on the base identity (A red, T blue, C green, G black) and another rule changes the label to `x`. In the second track, the wingding font (a font whose letters are symbols) is used with the same color scheme but now each label is changed to *n*, which is a square. In the third track, the label is *l*, which is a circle, and the color is adjusted to make C/T white, A red and G black.

Individual labels stacked in about 12 layers in Figure 30 tracks to avoid overlap. Because label density was constant (a label for each base position), and the symbol size for each label was the same, the labels fit neatly into a ring.

When a text track is used with glyphs to show non-uniformly spaced data, the glyphs will stack in a variety of patterns that reflects the density of the data. This is shown in Figure 32, which shows density of genes. Genes implicated in cancer are shown in *red*, genes in the OMIM database (which stores genes related to disease) are *orange*, and all other genes are *green*. The size of the glyph reflects the number of genes in a 1 Mb window.

We've already seen in Figure 30 that rules can be used to dynamically change the text label (where a base pair, e.g. T, was changed to an *X*, *n* or *l*). This method is leveraged in the gene density track. The input data uses the category of the gene (cancer, OMIM, other) as the label. The label size is already precomputed to reflect the gene density for that category.

A glyph track can be used to fill the entire image, as shown in Figure 31.

```
# gene density text track
...
hs1 111000000 111000000 cancer label_size=1p
hs1 115000000 115000000 cancer label_size=2p
...
hs1 100000000 100000000 omim label_size=7p
hs1 101000000 101000000 omim label_size=4p
...
hs1 100000000 100000000 other label_size=6p
hs1 101000000 101000000 other label_size=4p
...
```

Using rules, tracks are created to show data points with a specific label. The inner track in Figure 32 shows only data points with “other” label, the next track with “omim” and the outer red track inside the ideograms with “cancer”.

Circos Architecture

Implementation

Circos is written in Perl, which is available for nearly any computing platform and has been designed to be easily incorporated into genomic data analysis workflows. As a command-line application, controlled by plain-text configuration files, Circos image generation can be scripted and automated by wrapper programs or web forms. For an example of how Circos can be driven through a web interface, see the online tableviewer at *http://mkweb.bcgsc.ca/circos/tableviewer*.

Configuration

The configuration file has a hierarchical with parameters stored in blocks, which may be nested. Because of their simple format, configuration files can be easily created by other scripts to automate Circos. For example, the online tableviewer (*http://mkweb.bcgsc.ca/circos/tableviewer*) uses a configuration template which is adjusted based on input collected through a web form.

```
# circos.conf

# frequently changed parameters are found at the root of the configuration
karyotype   = karyotype.txt
chromosomes = chr1;chr2

# image block controls output format, image size, automatic transparency, etc
<image>
radius = 1000p
...
</image>

# position and thickness of ideograms, ideogram spacing and labels
<ideogram>
...
</ideogram>

# position and size of ticks, tick labels and grids
<ticks>
...
</ticks>

# data tracks such as histograms, scatter plots, heat maps, connectors, etc
<plots>
<plot>
type = histogram
file = histogram.txt
...
</plot>
<plot>
type = scatter
...
</plot>
</plots>

# links
<links>
<link chain>
ribbon = yes
file   = links1.txt
...
</link>
</links>

# highlights (these are drawn underneath all image elements)
<highlights>
<highlight>
file = highlight.txt
...
</highlight>
</highlights>

# house keeping definitions, such as colors and fonts

<colors>
red = 255,0,0
...
</colors>
<fonts>
default = fonts/arial.ttf
...
</fonts>
```

To keep the configuration modular, it is helpful to separate functional parts of the file into multiple files. This is additionally useful when parts of the file don't change (e.g. color, fonts). Importing content from other configuration files is done using the `<<include>>` parameter.

```
# circos.conf

# read ideogram configuration from another file
<<include ideogram.conf>>

# define ideogram configuration in ideogram.conf
# <ideogram>
# ...
# </ideogram>

# read tick configuration from another file
<<include ticks.conf>>
```

Practical Sessions

Session 1 – Ideogram Layout

In this first practical session you will learn how to position, order, crop and format the ideograms themselves. At the end of this session you will create the image shown in Figure 34, which will act as the template for the next session, in which data tracks will be added.

Session 2 – Data Tracks

In this session 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. Over the course of this session you will incrementally build an image that starts from the template from the previous session (Figure 34) and, by adding histograms, heat maps, tiles, links and highlights, ends in Figure 35.

Session 3 – Links and Rules

This session will focus on links and application of rules to links. You will create an image shown in Figure 36, which depicts the synteny between the mouse genome and human chromosome 1.

You will be introduced to two utility scripts that are included with Circos, `bundlelinks` and `binlinks`, which you will group links and create histogram density tracks.

Visualization Guidelines

In this supplementary section, I present methods of visualization design that improve legibility, clarity and help focus your message. Using examples from literature, we'll see how simple principles can be used in practice to make your visual presentation more effective and more attractive. I will include reinterpreted versions of the figure (Figure 41) to demonstrate how sometimes making minor changes (most frequently, *simplifying* changes) can have significant benefit.

Identify and Refine Your Message

It is much easier to create an effective visualization when you are clear about what you want to say. This may sound somewhat surprising – how can you create a figure without a message? Unfortunately, it happens all the time.

It is common to see a figure that shows all the data (often too much data), leaving the reader to fend for themselves. This is usually due to the complexity of the data set and in such cases the authors have likely not chosen a suitable representation for the figure.

The examples below show published figures that lack a clear message. Many such figures can be very attractive (Figure 37) but they do not communicate the information better than a well-written sentence.

Using a visual representation that is suitable for algorithmic traversal does not always create a useful visual representation. Figures of large networks or graphs (Figure 37, Figure 38, Figure 39) are typically hard to interpret. When visualizing a data structure ask yourself whether the native form of the data is the best vehicle for communication.

When should you show the entire data set? If the data contain an emergent pattern, such as in Figure 40, then you can be justified in doing so. This design is a great example of a pattern that is very difficult to parametrize (how do you curve fit a human form?) but very easy to recognize.

Creating figures with even simple messages require care. The comparison of ventricle sizes in Figure 42 requires that the reader be able to distinguish small changes in the size of an annulus. To ensure that the changes are communicated without ambiguity, exaggerating them (even beyond realistic limits of the underlying phenomena) can be very effective in removing doubt.

Respect Limits of Output Resolution and Visual Acuity

You must always respect both the resolution limit of the output medium (screen, printer) and the limit of visual acuity of the reader. Commonly journals require that line art be prepared at 1200 dpi and bitmaps at 300dpi. Both of these resolutions are significantly higher than the acuity of the human eye, which can only separate two objects separated by about 1pt at reading distance (1pt = 1/72 inch = 0.35mm).

The practical resolution limit should therefore be chosen with visual acuity in mind. This is illustrated in Figure 43, where the 1pt limit is translated into genomic distance, given the size of the sequence that spans a printed page.

It turns out that both print and screen resolution (3 pixel viewing limit for 24" screen at 1,920 horizontal resolution) limits suggest that no more than 600 divisions be created on a scale.

This resolution limitation makes the display of sparse genomic data difficult. Views of the type shown in Figure 44 are common in the literature and many are not easily interpretable. To be fair, the authors do not have a lot of options in displaying the data in a printed form, if they seek to maintain a physical scale.

Figure 45 is an example of what happens when the complexity of the data outweighs the number of distinguishable pixels. This figure suffers from the additional problem that the data appear to lack a pattern. When you are creating visualizations, make sure that your end product can be distinguished from a randomly generated data set!

Choose an Appropriate Representation

As mentioned above, the native representation of the data (e.g. a network) may translate poorly into a figure designed to be parsed by a human reader.

When faced with a multi-dimensional data set, it is not always easy to discover an single intuitive and interpretable visual representation. The authors of Figure 46 attempted to show this kind of complex data set as a pair of Venn diagrams. Note that within each overlapping region data are further subdivided by an independent category (pathway). The entire representation is confounded by the presence of the filter Venn circle, which by definition, has no data points that are unique to it. The figure is so complex, that counts within each region are explicitly provided, obviating the need for the data circle glyphs.

The data in this example is too complex to present in a single figure. Instead, the redesign shows the data presented as a two-part figure. The first panel presents the color scheme and top-level statistics of the two algorithms and the result of applying the filter. This panel acts as an entry point into the data, presenting enough information to provide the reader with firm grounding, but not too much as to overwhelm them. The second panel shows fine texture in the data, by presenting it categorized by pathway. Since the reader has already parsed the first panel, they arrive at the second panel with prior knowledge (color scheme and filter condition) and therefore can quickly navigate within the second panel.

In contrast to the complexity of Figure 46, an example of mishandled simplicity is shown in Figure 47. Whereas the Venn diagram was *too simple* to handle the complexity of the data set in Figure 46, it is *too complex* to handle the simplicity of the data set in Figure 47. These two example nicely frame the visual load limit of a representation.

Remove Excess Ink

An important property of a visualization is the data-to-ink ratio. How much ink has been used for data, and how much for all the ancillary information, such as navigation aids, borders, legend and so on. Visualizations that are information rich and have a high data-to-ink ratio are generally successful (although there's a limit to how much a human reader can parse), and figures which use too much ink simply confuse the reader and hide your results.

A good example of information hiding due to excess ink is shown in Figure 48. Much of the same information is repeated, making it difficult to distinguish where the different amino acid codes are located (some letters look more different than others – while A and V are visually distinct, R and B less so). The authors use ink (borders) to break each sequence into segments of three amino acids. Unfortunately, this adds so much visual weight to the figure that the data loses its presence.

A lot of ink can be removed in Figure 48 by (a) showing the same information only once and (b) using space to break up the sequence. By showing the consensus sequence above the alignment panel, changes can be quickly spotted. Note that "." is used to indicate no change to provide a horizontal guide for each protein.

Excess ink also impacts the effectiveness of Figure 49, though in a different way. In the alignment in Figure 48, too much ink was used for grouping and duplicated information. In Figure 49, excess ink is used to provide physical context to the data. The authors chose to show the entire chromosome to display information about the telomeres. This results in a figure in which nearly all of the ink is spent on the ideograms, which themselves do not add any value to the image (is the reader an expert cytogeneticist who can natively navigate around the banding pattern?). By removing the ideograms from the figure, the information can be presented in a tidy pair of rows.

The use of excess ink is a common problem in figures, and none less in splicing diagrams. The main information in a splicing diagram relates to the splicing (where a splice event is simply an association between two exon ends), not the exons themselves. In Figure 50 so much ink is used on the exons (many of which vary in size and position) that any pattern in the splicing events is impossible to discern. The figure also suffers from excessive redundancy in labeling. The redesign offers a way to organize the information such that the splicing events are prominent.

The problem with excess ink arises in figures which use ink to communicate an absence of information. In Figure 51 the data set is binary (sample is/isn't present in pool). However, both states are encoded by ink (0 and 1) – in fact the 0 uses more ink than the 1, but the 1 state is more important! The visualization is swamped by too much emphasis on absence of data (or more specifically in this state, incorrect state encoding).

Avoid using "0" or "none" or even "-" to show that data is missing. Just leave the region of the figure blank. Sometimes it is clear that a blank region corresponds to an "off" state or missing data, such as in the redesign in Figure 51. If you need to distinguish between three states, such as *yes*, *maybe* and *no* consider using ● (*yes*), ○ (maybe) and a blank for *no*. These two glyphs naturally encode a hierarchy (○ is perceived as less than ●).

Avoid excess ornamentation of the kind shown in Figure 52. In an effort to make the figure look more visually exciting, the authors have sacrificed legibility. Keep the visualization simple or, at the very least, make sure that your message remains intact during any prettification.

In Figure 51 we saw how encoding both states in a binary variable can impact clarity. In Figure 53 a similar problem appears: too much ink is used to encode the wrong part of the figure. Since it is the relative position of nucleosomes that is important, it is not necessary to attempt to draw the nucleosomes realistically.

One of the ways in which ink can be used excessively is in the repetition of axes and tick labels in a multi-panel plot. In Figure 54 a large number of scatter plots is shown, each framed by a border. The border is thick and, in addition to the lack of alignment of labels, overwhelms the data. By removing the axes, while still maintaining a clear depiction of scale, the data is given prominence in the redesign.

Avoid Redundancy

The alignment in Figure 48 and splicing diagram in Figure 50 both suffered from redundancy. In both cases, text was simplified by factoring out the consensus sequence (in the alignment) or decomposing the labels into independent categories (in the splicing diagram). It is very common to see figures with repeated text, and this is illustrated in Figure 55, where the example is striking because the rest of the figure is perfectly acceptable. When all data points share the same part of a text label, this part should be factored out. Here the authors use both color and text to distinguish between *mu-*, *hu-*, *fu-* and *zf-* classes of labels. Avoid using two parameters (e.g. color and shape, shape and size, color and size) to encode the same piece of information. The reader does not expect this – in fact they expect that the two parameters are independent – and it takes a while to determine that they are, in fact, dependent.

Provide a Visual Framework – Use Grids

You may have heard about the use of grids in layout and graphic design. One look at a newspaper illustrates this principle, in which information is structure around an underlying grid pattern. In a news paper, all columns on a page have their left edge at a small number of vertical guides. Moreover, the position of these guides is repeated across the entire newspaper. While such a grid system may sound unexciting to you, and without visual *flare*, ask yourself whether it is indeed flare that is required, or simply effective communication?

The reason why grids work is because they lower the burden on the eye as it scans content on the page. If there is a large number of possible points at which the eye should stop (exon start positions in Figure 50 unnecessarily vary), the reader will fatigue easily. Same is the consequence if alignment is not uniform (note that distance between adjacent columns of 0/1s in Figure 51 is not constant). Notice how aligning the labels in redesign of Figure 55 helps legibility.

To illustrate how a grid system can work, the redesign of Figure 56 applies a grid layout to the pie charts. With the grid in place, a pie chart cannot appear anywhere in the figure, but only at a

fixed number of positions. The grid should be fine enough to accommodate positioning requirements but not too fine as to lose its effect. By placing the pie charts on a grid and limiting the range of relative positioning of the pie charts with respect to their regions, the association between the pie charts and regions is made clear without many link lines. In fact, only two link lines are required for regions where the charts could not be unambiguously placed closer to their regions.

In the redesign of Figure 56 I have used a light solid shade for the land mass of India. This may first appear to contradict the requirement that minimal ink be used. While it is true that as little ink as possible should be used, ink should not be removed at the expense of clarity. In the case of a map, it is easier to distinguish the land mass from surrounding areas when the land mass is shown as a solid color.

Choose Colors Based on Perceptual Characteristics

The perceptual characteristics of color should be taken into account when selecting a color scheme for data encoding. Up to now, you may not have thought about the distinction between color *characterization* and *perception*. The difference is illustrated in Figure 57.

The consequences of *perception* vs *characterization* of color is not merely academic. Although the details of color science are complicated, relatively straightforward principles can be applied to color choices to leverage our knowledge about perception. For example, because yellow is perceived to be the brightest color, the yellow bars in Figure 58 draw attention away from the other two categories. In fact, you will find it hard to focus on the darker bars – your eye naturally fixates on the yellow ones. By adjusting the colors to have the same luminance (perceived brightness), the redesigned version is more harmonious and no single category stands out.

An explicit demonstration of luminosity difference in pure colors is shown in Figure 59. In the first row colors differ only by their hue. Even though their *value* in HSV space is the same (maximum), some colors appear brighter than others (yellow *vs* red). This discrepancy between the difference in perception for constant parametrization is due to the fact that HSV color space is not *perceptually uniform*. Thus, real benefit can be derived by simply requiring constant luminance for all colors in a scheme.

As an example, let's apply luminance normalization to the default human chromosome color assignment used by UCSC browser. This color palette is shown in Figure 60. Color coding is required for conservation and synteny tracks, which associate a chromosome with each position – this encoding is done by color.

You'll notice that the yellow for chromosome 10 immediately jumps out, as does the bright blue of chromosome 17. Indeed, these are some of the colors with the highest luminosity. The palette is not perceptually uniform with respect to brightness and this confounds visual interpretation because the eye is preferentially drawn to certain colors.

The table below assigns each chromosome to one of four luminance ranges. 9 chromosomes have bright color assignments, and of these 9, 5 have very bright colors (luminance > 90).

lum	n	chromosomes
0 – 25	2	2 14
26 – 50	5	1 4 13 19 22
51 – 75	8	3 5 6 8 15 20 21 X
76 – 100	9	7 9 10 11 12 16 17 18 Y

```
lum > 90 is underlined
```

An alternative assignment is proposed in Figure 61, which shows three color schemes for which luminance has been normalized to 70, 80 and 90, respectively. These normalized colors are defined in `sessions/etc/brewer.conf` as `lum70chr*`, `lum80chr*` and `lum90chr*`.

When faced with having to choose many colors for a scheme, perceptual uniformity of colors becomes more challenging to maintain through manual design. While the three histogram colors in the redesign in Figure 58 can be chosen by eye (up to constant luminosity, which should be fixed numerically), it is much harder to construct a 10-color scheme, as required in Figure 62.

Luckily, such palettes are available courtesy of Cynthia Brewer, who painstakingly created harmonious schemes for qualitative, diverging and sequential palettes. These palettes are called Brewer palettes and you can learn more at *www.colorbrewer.org*.

Examples of several Brewer palettes are shown in Figure 63. The generation of these palettes has been generalized and implemented in *PaletteView*, which attempts to capture Brewer's decisions algorithmically. For more details about this scheme, see

Wijffelaars M, Vliegen R, van Wijk JJ, van der Linden E-J. 2008. Generating Color Palettes using Intuitive Parameters. In Eurographics/IEEE-VGTC Symposium on Visualization 2008, Vol 27 (ed. A Vilanova, A Telea, G Scheuermann, T Moller).

Finally, avoid using a dark background for visualizations that contain fine detail. A striking example of how the background can hide data is shown in Figure 64. Simply by inverting the figure, data appears!

Where possible, avoid using a grey background (default setting in some versions of Excel), since it merely reduces the contrast of the figure. In fact, unless there is a compelling reason not to, always use with a white background.

Visualization Checklist

It is not possible to capture a global recipe for a successful visualization. Instead, figure design should be guided by investigating the data (density, sparseness, relationships), identifying aspects that are important and by maintaining respect for the readers visual and cognitive limits.

Remember, do not let your data and figures speak for themselves – you don't know what they'll say. Make sure that you help guide your readers to important patterns and appropriate conclusions. When creating a figure, ask yourself the following questions. As long as you are being honest in your answers, your figures should benefit!

Q What are the major questions that the figure should help the reader answer?

Q What are you trying to communicate? Does the figure communicate it clearly?

Q Is it clear to the reader where they should look?

Q Is a graphical representation really necessary? Does the legend obviate the figure?

Q Are there extraneous or ornamental elements? What can you safely remove?

Q Have I left the reader wanting more, or less?

□

Figures

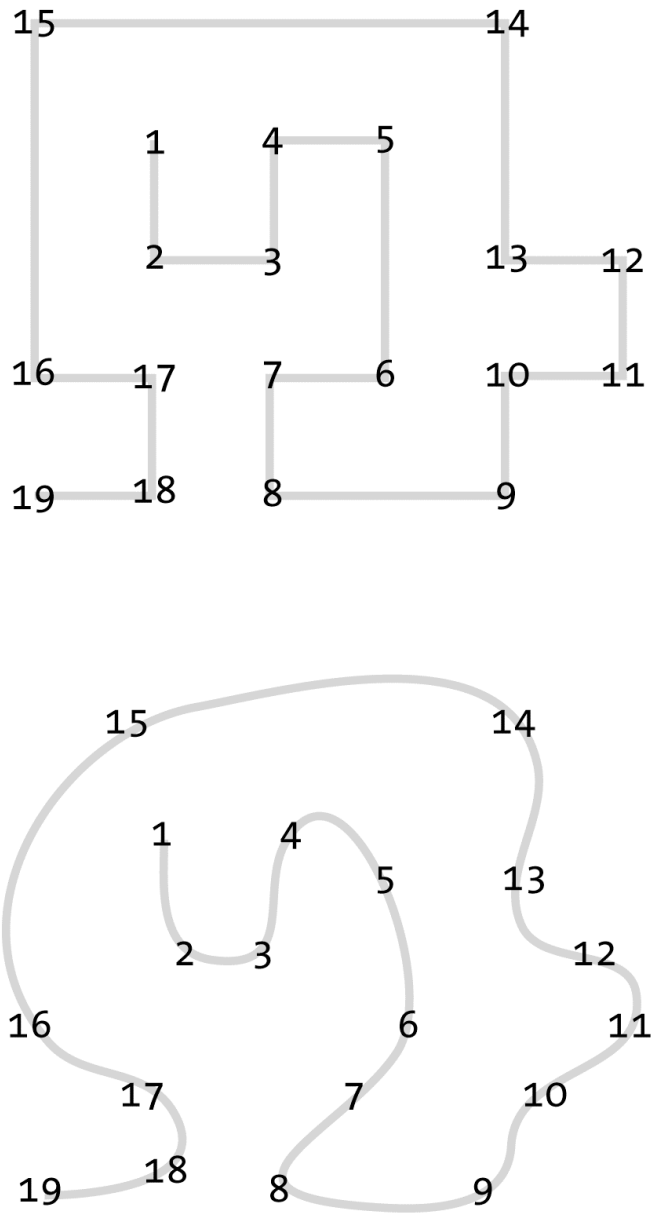


FIGURE 1

Curved objects are easier to visually follow.

Time yourself to see how long it takes you to scan through the numbers in the two shapes. You will find that effort in interpreting the top shape is perceived higher than the bottom shape.

Right angles in the top shape require more energy to traverse – you may find that switching eye movement from vertical immediately to horizontal is uncomfortable.

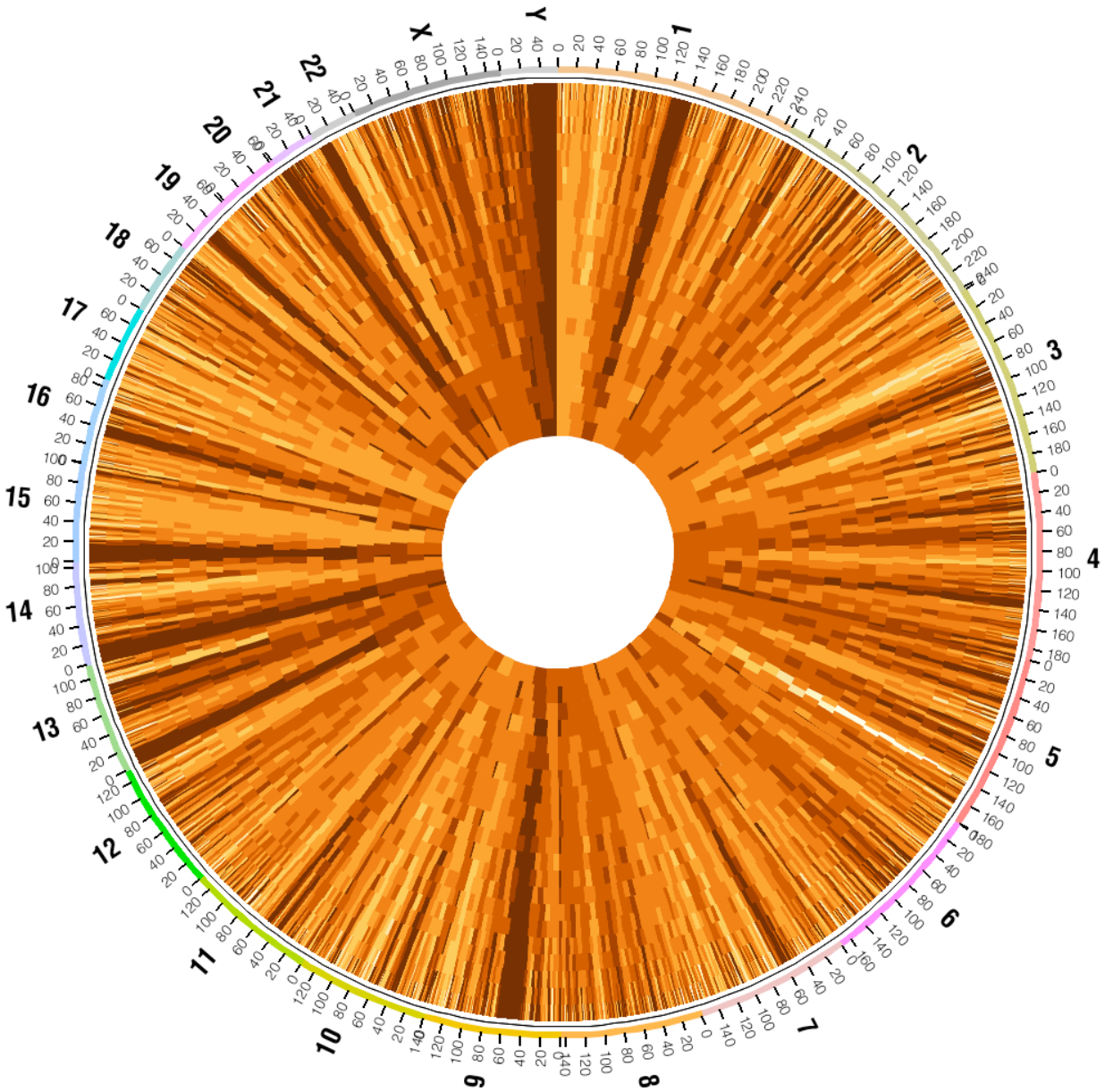


FIGURE 2

Gene density across the human genome shown in 21 concentric tracks at varying resolution.

In the inner track, resolution is shown across 50Mb bins and in the outer track in 1Mb bins.

The circular form naturally supports a range of resolutions, since the circumference of a track is proportional to its radial position.

`sessions/1/2`

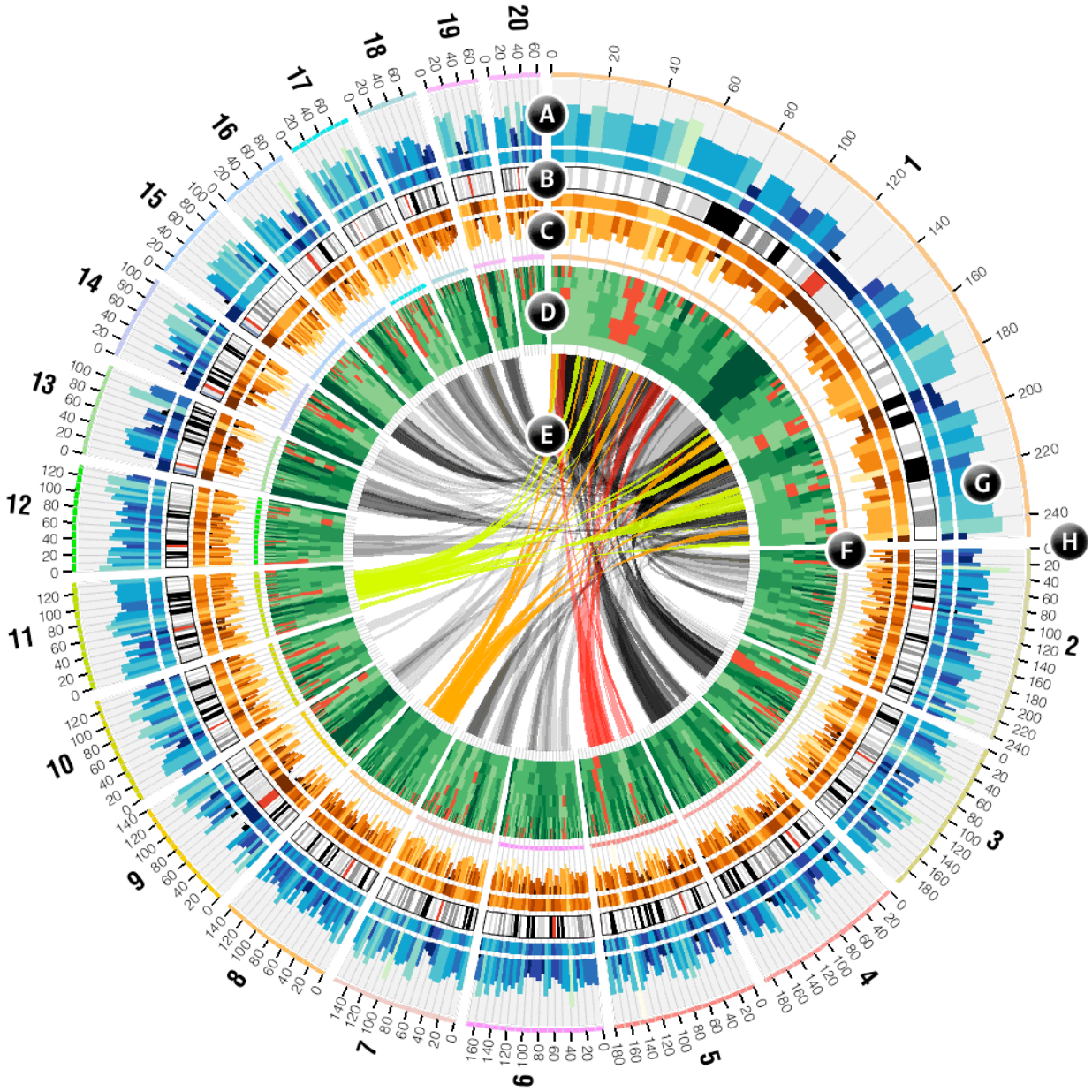


FIGURE 3

Ideograms are arranged circularly (B), and may have their scale altered (G, chr1 at *4x* magnification) with ticks and labels (H) placed anywhere in the figure. Data tracks placed either outside (A) or inside (C, D, E) the ideograms. Highlights are a special track type which is drawn underneath the grid (F), useful for color indexing.

Rules can be applied to any track to change the format of the data points, including color, geometry and visibility. Rules were applied to tracks (A,C) to color bins by height, (D) to color heat map by value and (E) to color links by size and position.

`sessions/1/2`

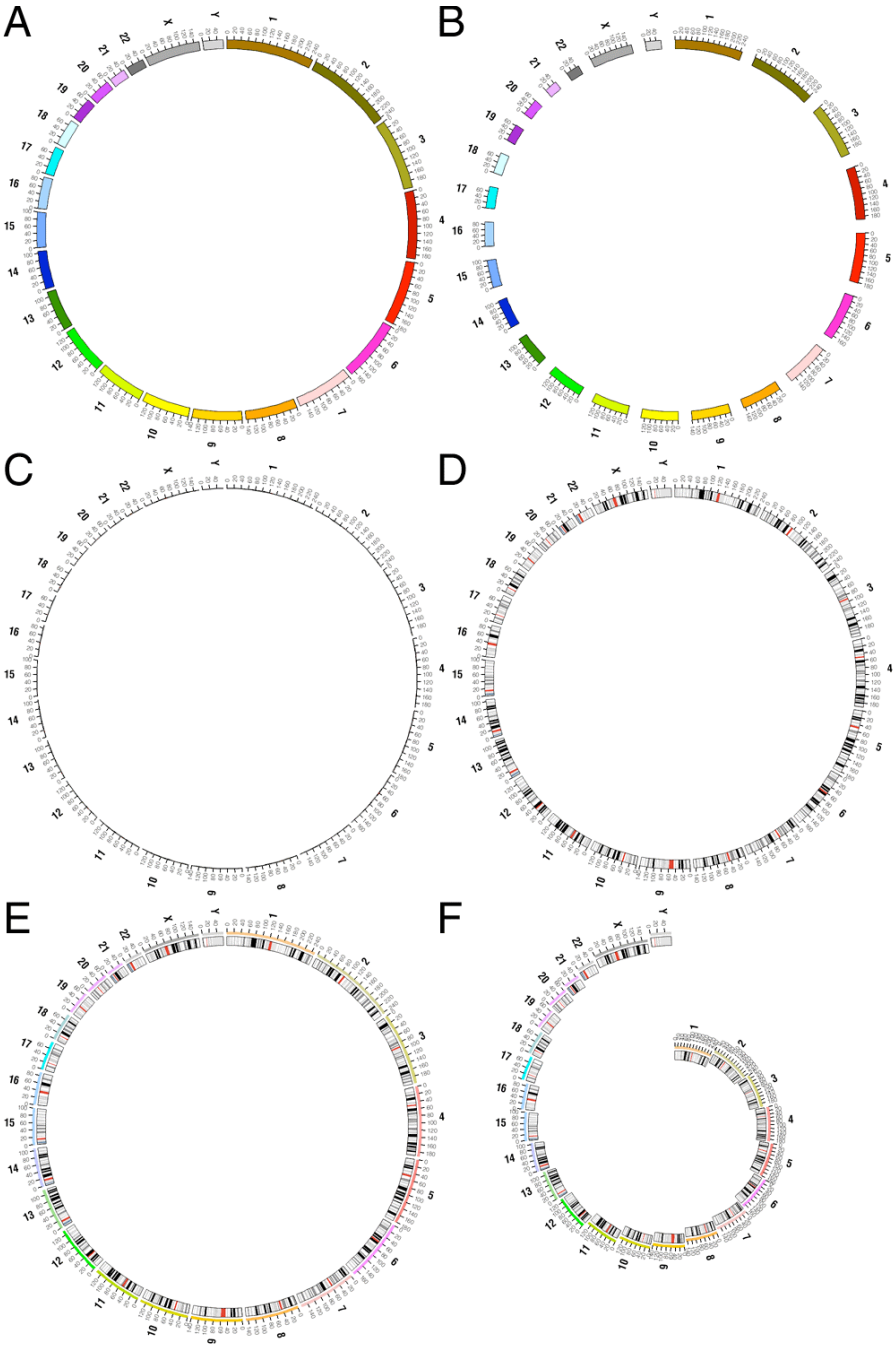


FIGURE 4

Ideogram layout is flexible. Chromosomes can be assigned a color, which is used to format the ideogram (A). Spacing between ideograms can be adjusted globally or independently for each pair (B). Thickness of ideograms can be altered to reduce clutter (C). Chromosome definitions can include regions corresponding to cytogenetic bands, which are depicted as as colored blocks within the ideogram (D). Highlights can be added to provide a color index (E). Radial position of each ideogram can be independently adjusted (F).

`sessions/1/3`

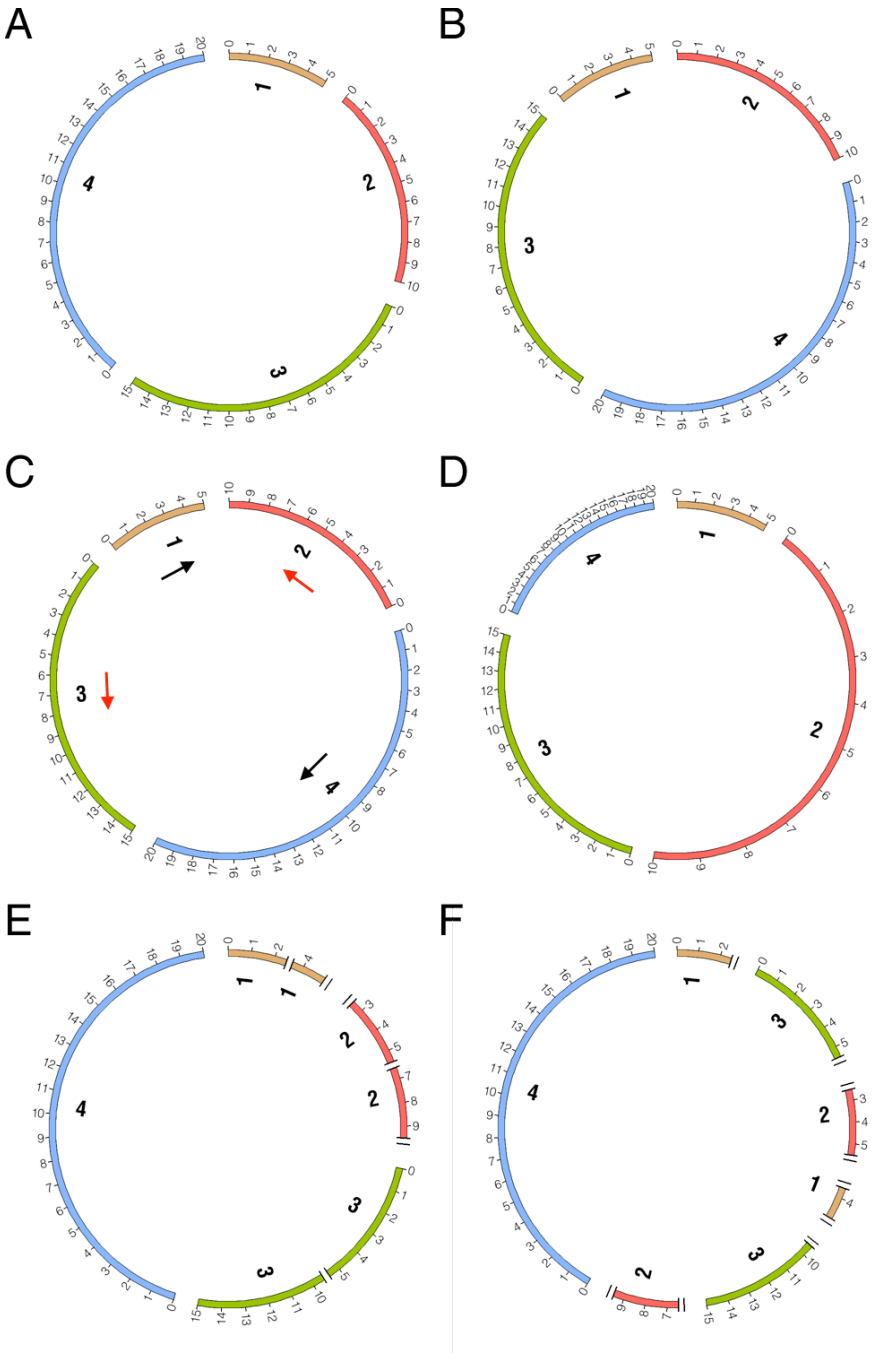


FIGURE 5

Ideogram order, scale and axis breaks can be freely mixed. (A) Four chromosomes 5, 10, 15 and 20Mb in size are (B) rearranged. (C) Orientation of scale for chromosomes 2 and 3 are reversed. (D) Scale of chromosome 2 has been adjusted to 2.5x magnification and of chromosome 4 to 2x reduction. (E) Regions from chromosomes have been removed from the figure (chr1:2.5-3.5Mb, chr1:4.5-5, chr2:0-2.5Mb, chr2:5.5-6.5Mb, chr2:9.5-10Mb; chr3:5.5-9.5Mb), introducting axis breaks. (F) Order of individual regions from previous panel has been rearranged.

`sessions/1/4`

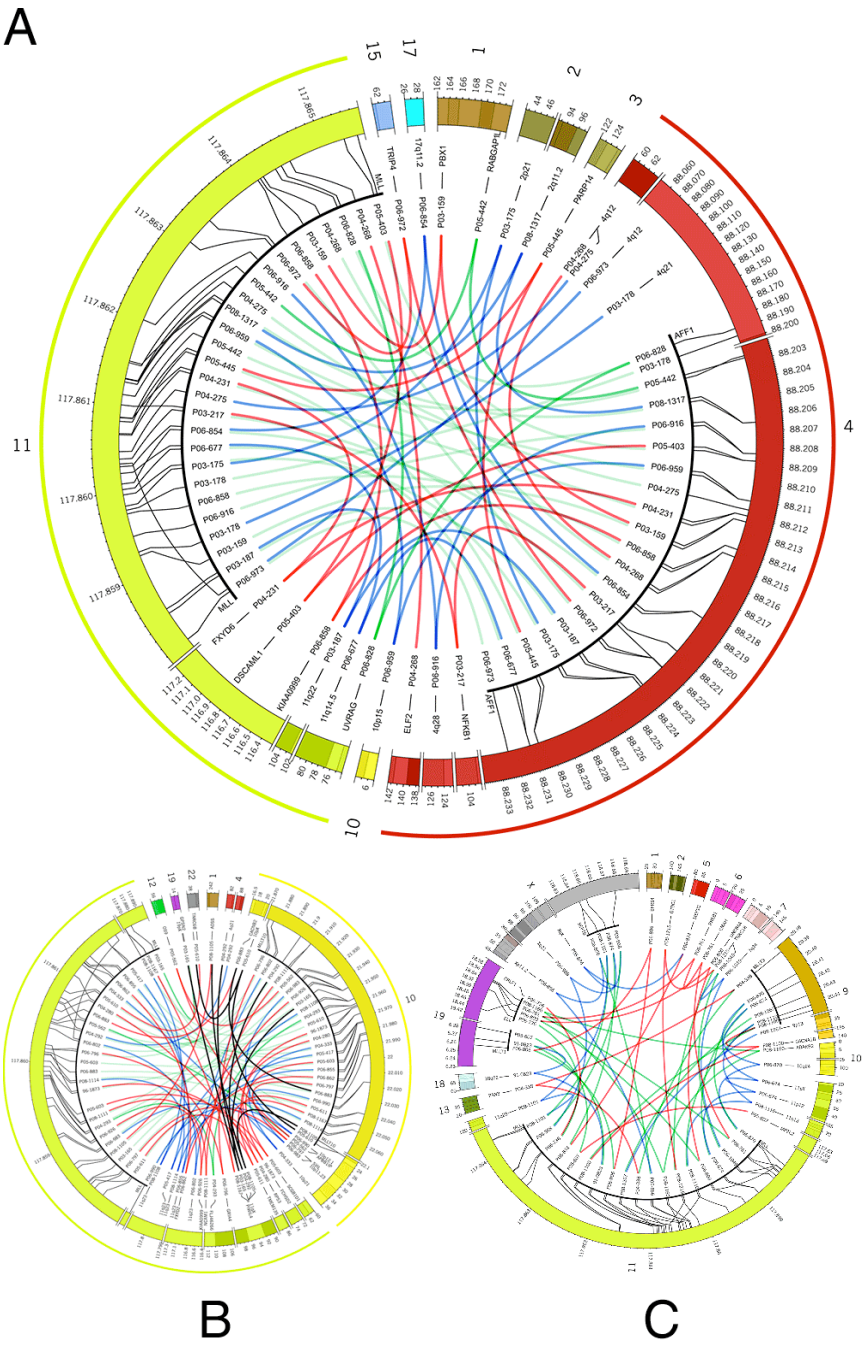


FIGURE 6

The most frequent complex rearrangements involving *MLL* and (a) *AFF1/AF4*, (b) *MLLT10/AF10* and (c) *MLLT3/AF9, SEPT6, MLLT1/ENL*, *ELL and TNRC18*. Localization of chromosomal breakpoints and UPN of individual patients are indicated. Colored lines: green lines: in-frame fusions; red lines: out-of-frame fusions; blue lines: no partner gene present at the recombination site.

Meyer, C., E. Kowarz, et al. (2009). "New insights to the MLL recombinome of acute leukemias." Leukemia **23**(8): 1490-1499. Figure by M Krzywinski.

`sessions/1/5`

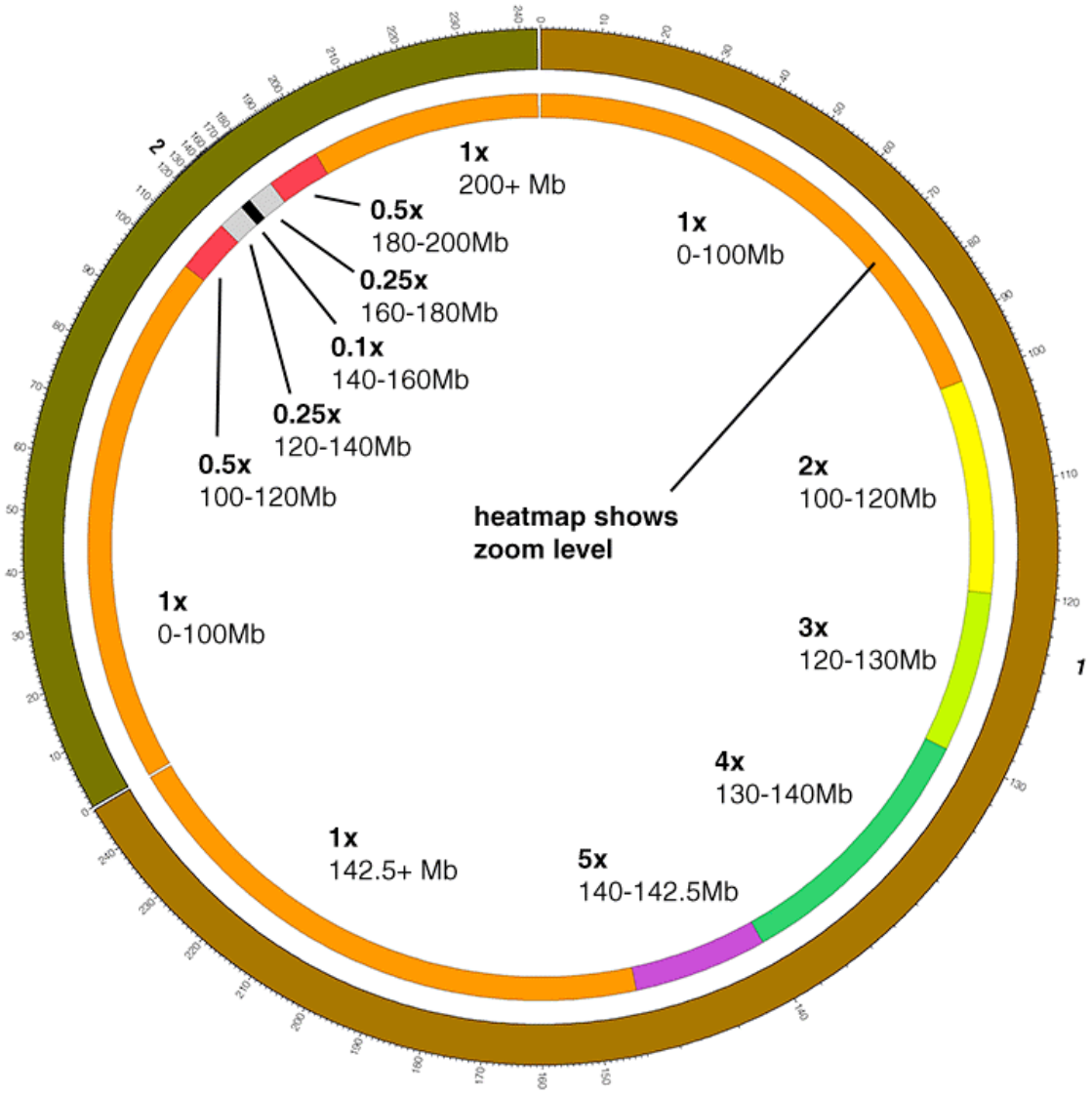


FIGURE 7

The scale within an ideogram can be freely adjusted from reduction to magnification. Here, zoom regions are defined within the ideograms of chromosomes 1 and 2 to affect variable reduction and magnification of regions of these chromosomes. For example, chr1:140-142.5Mb is shown at *5x*, whereas chr2:140-160Mb is shown at *0.1x*.

`sessions/1/6`

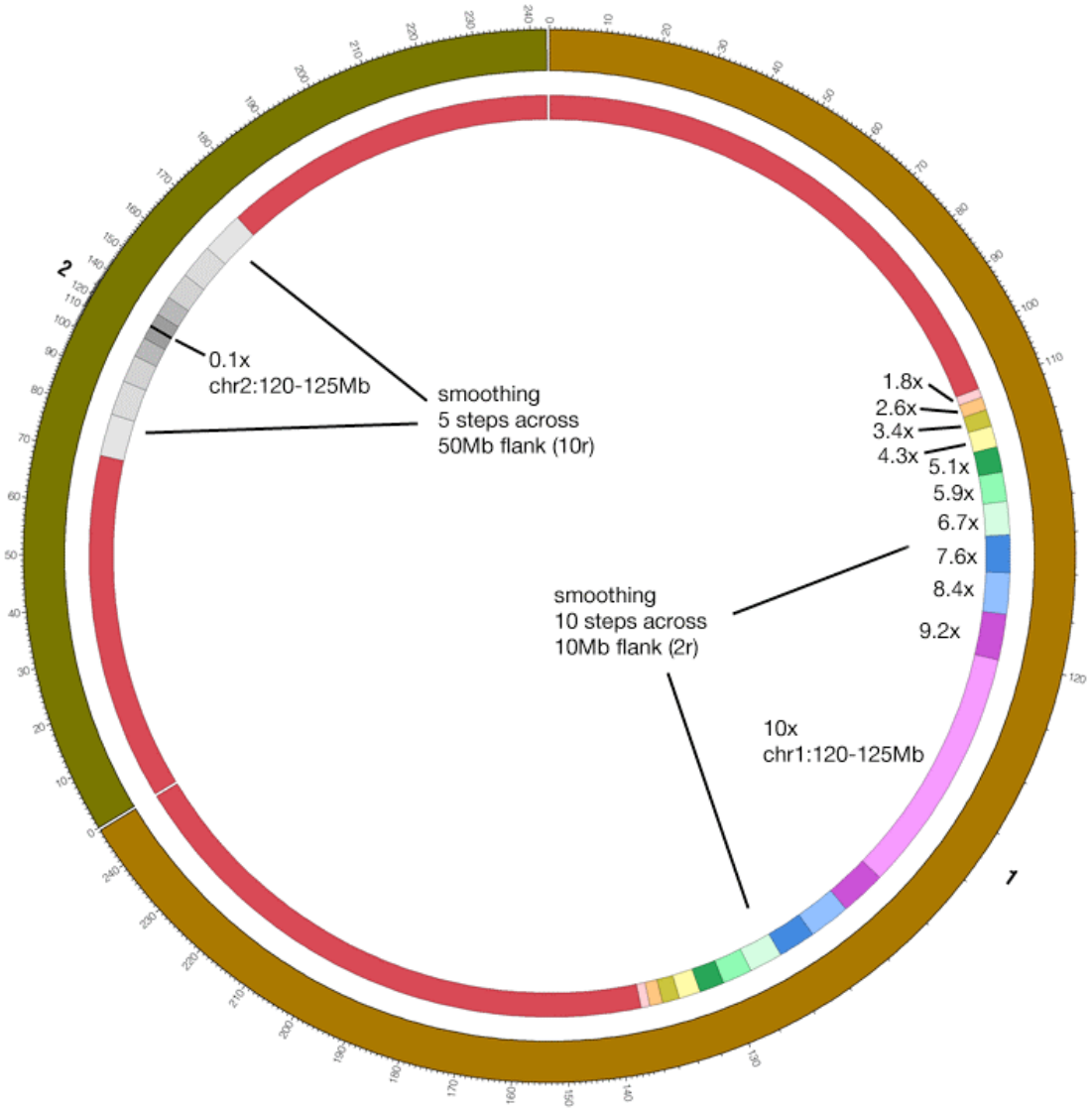


FIGURE 8

Using continuous scale change, a region of magnification (chr1:120-125Mb *10x*) or reduction (chr2:120-125Mb *0.1x*) is automatically used to influence the scale in its immediate vicinity to create a smooth scale transition.

`sessions/1/7`

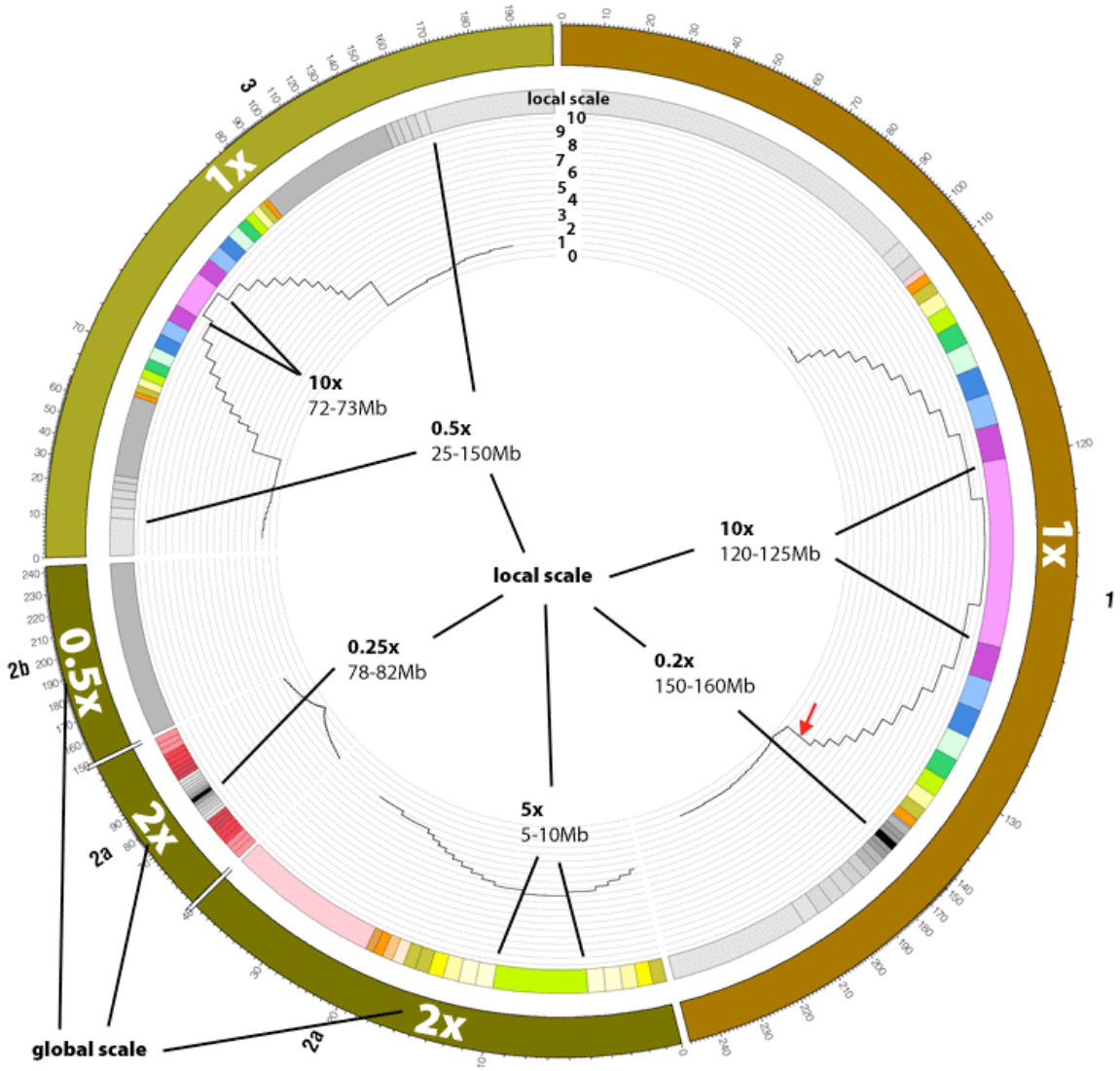


FIGURE 9

Using continuous scale change, a region of magnification (chr1:120-125Mb *10x*) or reduction (chr2:120-125Mb *0.1x*) is automatically used to influence the scale in its immediate vicinity to create a smooth scale transition.

`sessions/1/8`

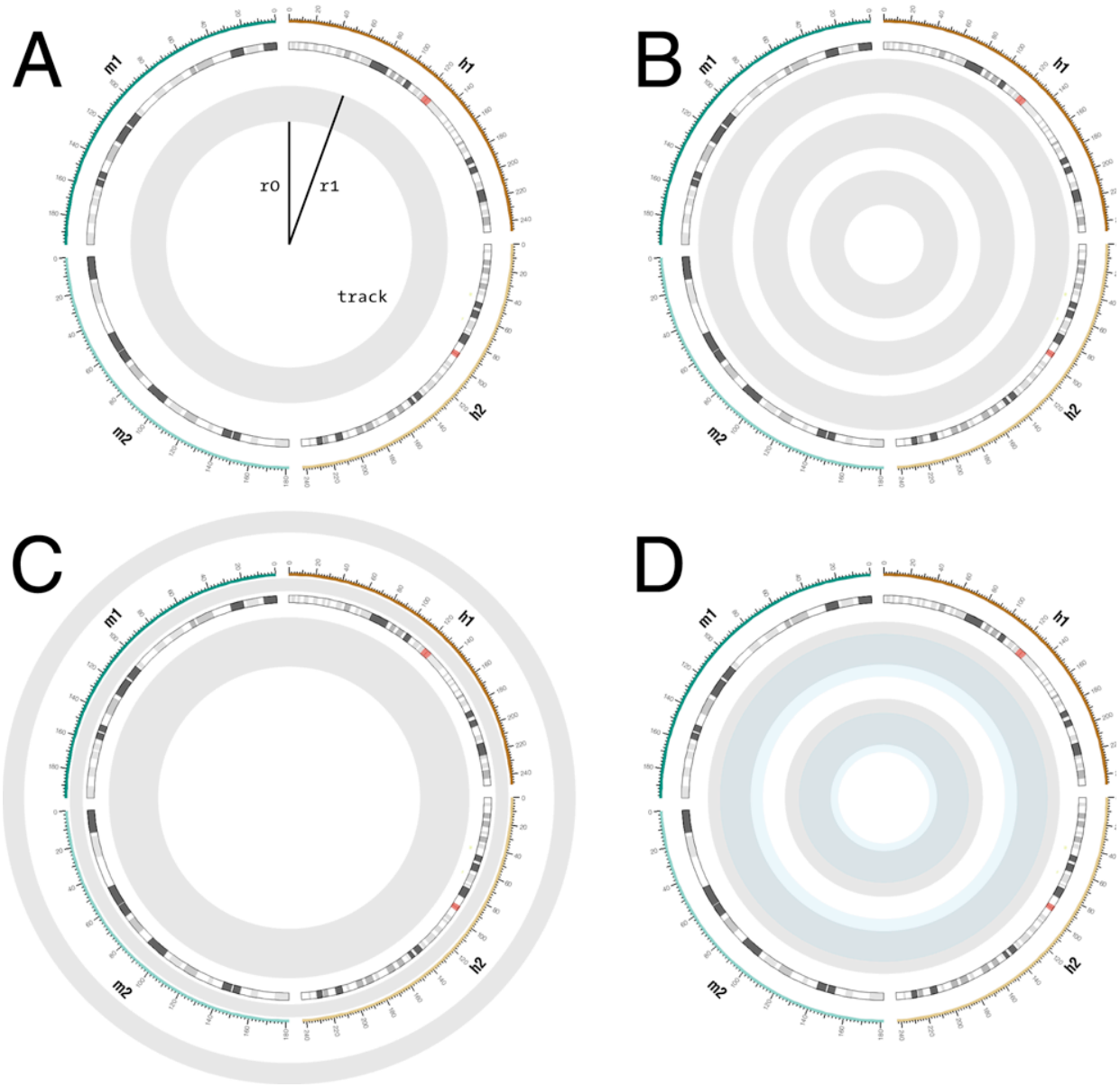


FIGURE 10

(A) Each data track confined to an annulus bounded by radii `r0` and `r1`. (B) Any number of tracks can be placed on the figure, and (C) at any radial position, including inside/outside ideogram circle and inside/outside ticks. (D) Tracks can be made to overlap and can be drawn in any order.

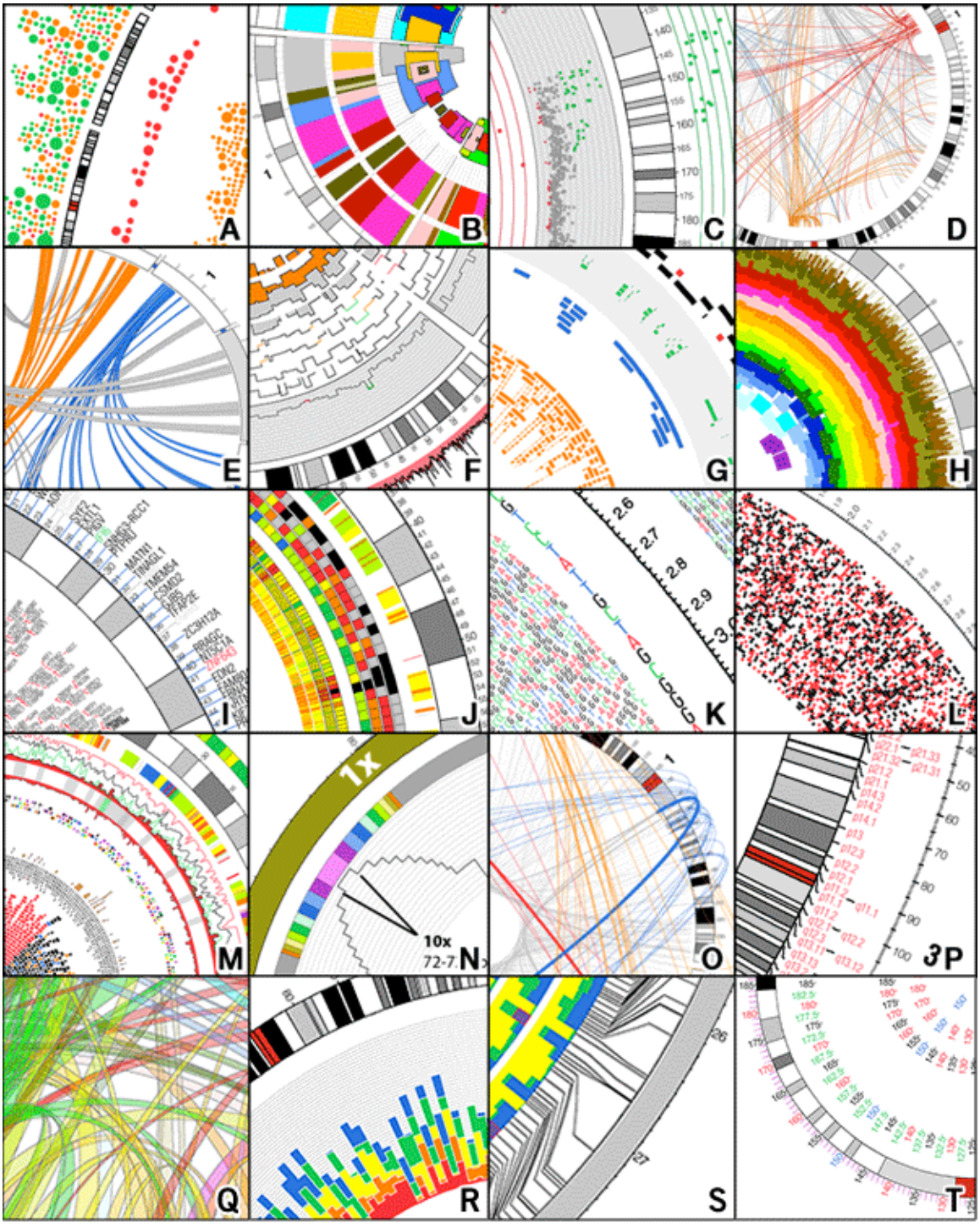


FIGURE 11

Some examples of Circos tracks and features. (A) glyph (B) highlight with depth control (C) scatter (D) paired-location (E) ribbon (F) histogram (G) tile (H) highlight with auto depth (I) text with auto arrange (J) heat map (K) high-density text (L) high-density glyph (M) multi-type composite (N) variable scale control (O) fine geometry control (P) flexible text and element placement (Q) transparent ribbons (R) stacked histogram (S) connectors (T) tick rings.

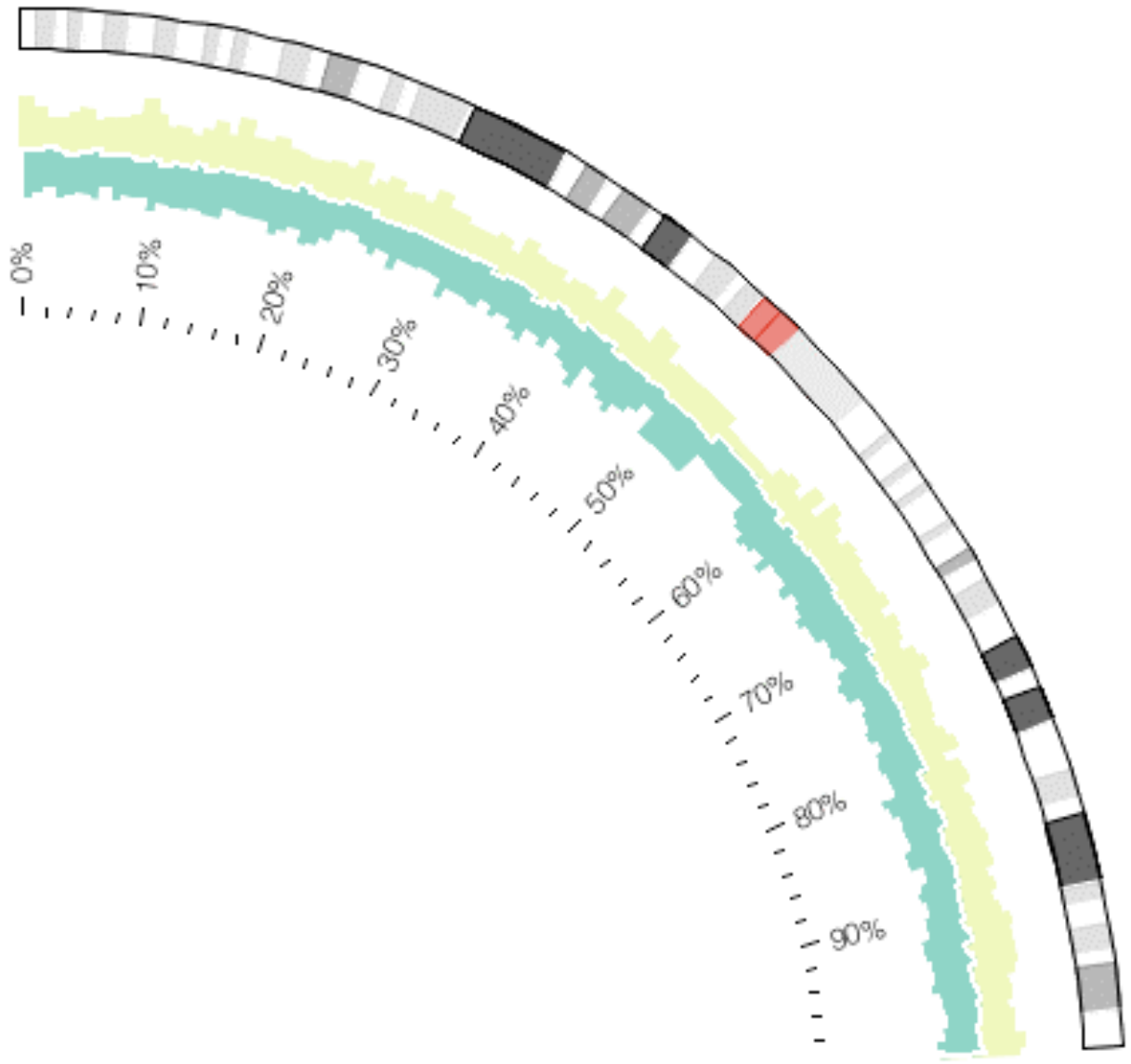


FIGURE 12

By defining three histogram tracks within the same radial region, and drawing the data in a specific order, a compound track can be created. In this example, three histograms were used to generate the final plot.

`sessions/3/2`

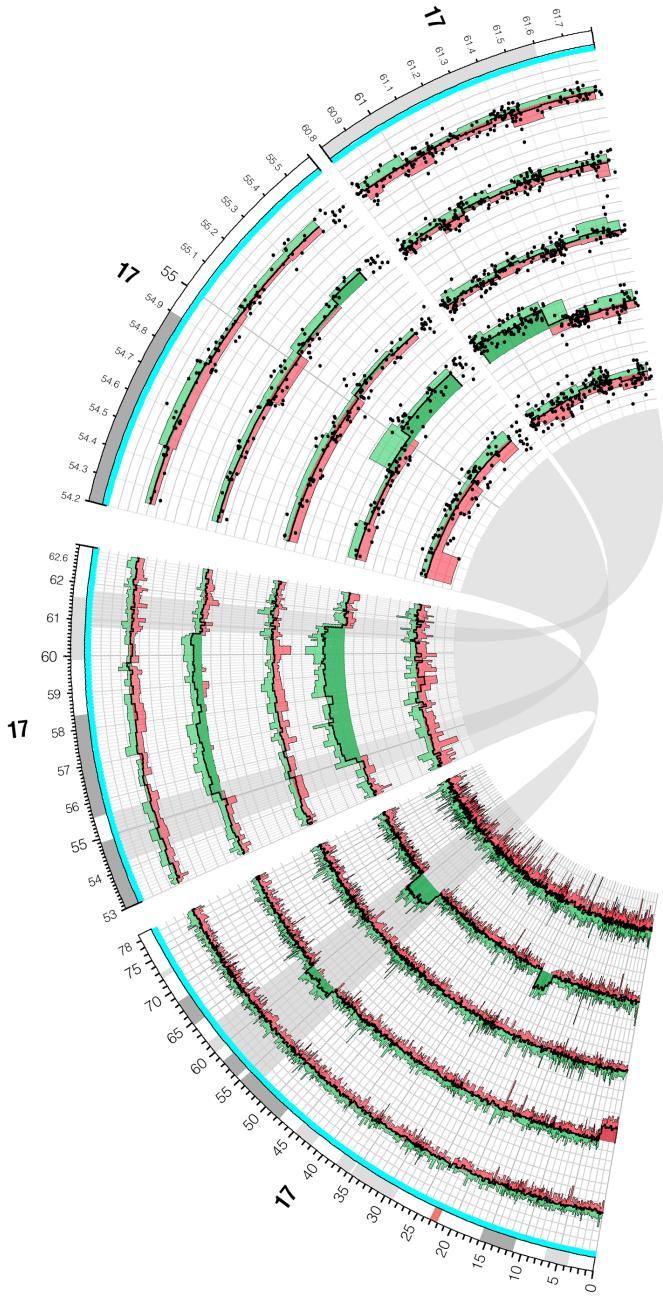


FIGURE 13

Various types of data tracks can be stacked. Here, two histograms, a line plot and a scatter plot are used to form a compound track. Five instances of this track are shown in the figure (each represents an individual biological sample). Using links and highlights, attention is drawn to the progression of scale increase within chr17:53-63Mb. This region is magnified at *5x* and smaller subregions are further magnified to *40x*.

Krzywinski, M., J. Schein, et al. (2009). "Circos: an information aesthetic for comparative genomics." Genome Res **19**(9): 1639-1645.

`sessions/1/9`

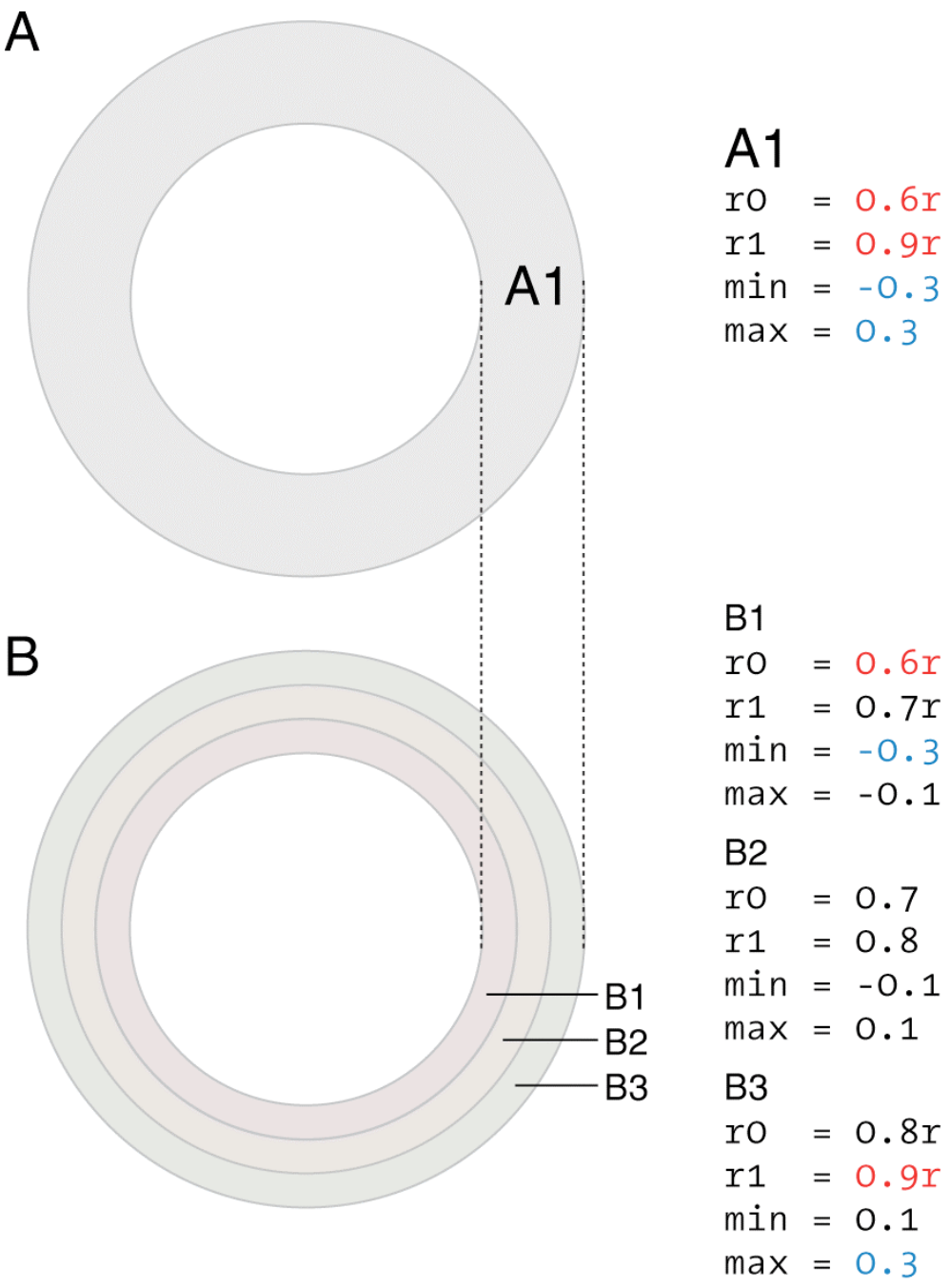


FIGURE 14

To apply different format values to parts of the same histogram bin, a single track (A1) can be partitioned into three (B1, B2, B3) (or more). The partitions occupy the same region within the figure and the same data value range and each use the same input file. Within each partition, histogram bins will be clipped to the partition data range. These bin regions will be formatted based on the partition's settings, allowing for a single bin to be built up from multiple and independently colored components.

FIGURE 15

The effect of partitioning a histogram track into three tracks. The histogram partitions (inside to outside) are given different background colors (red, orange, green) and the fill color for bins is also different for each partition. The histogram baseline is in the middle track. Bins within this track are orange and, if they extend outside of the range of this track, are clipped by the track'sfs baseline or top. These bins continue in adjacent tracks, now colored based on that tracks' formatting.

`sessions/1/10`

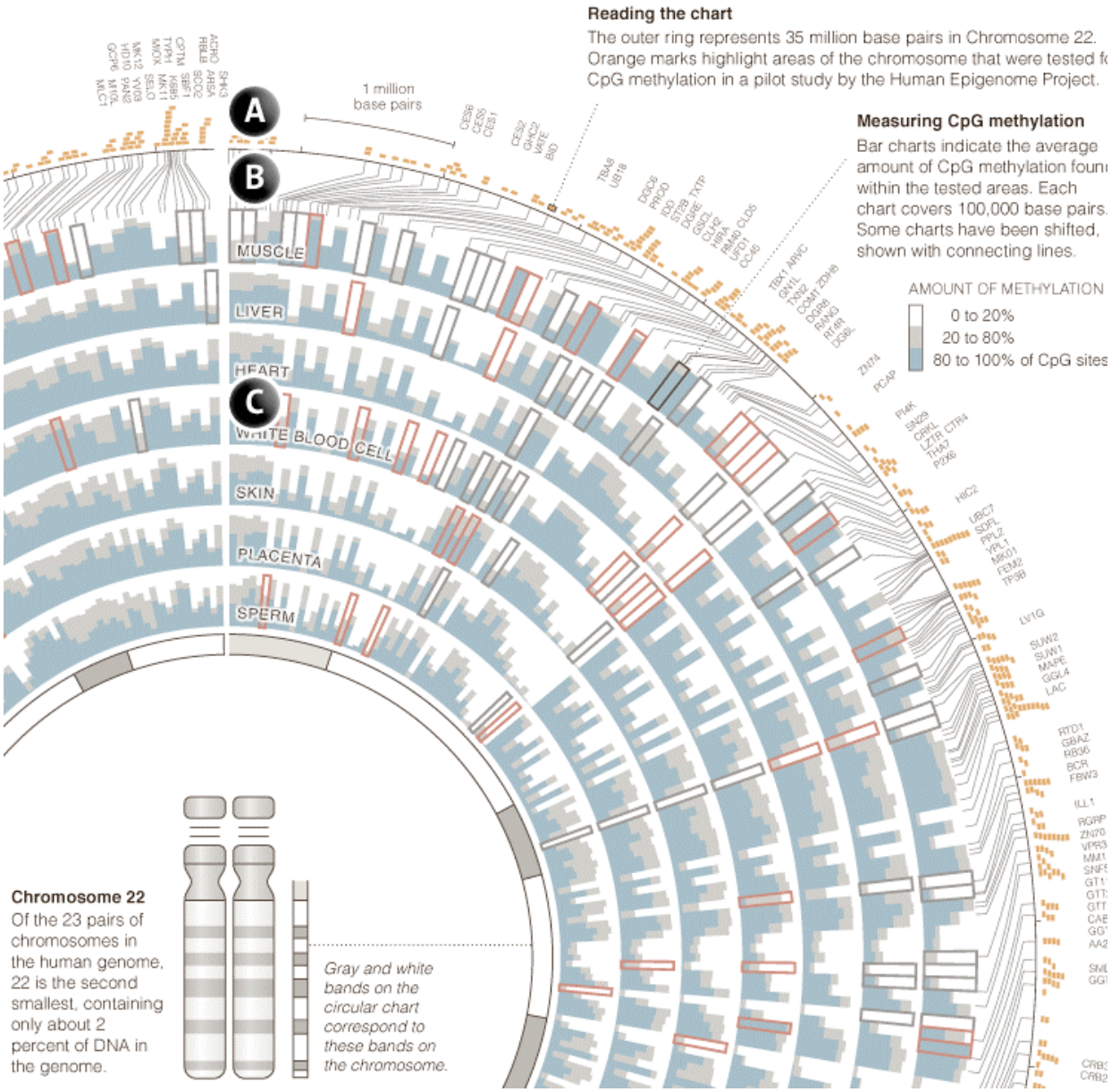


FIGURE 16

Data sets which do not sample the genome uniformly (A) can be effectively shown by remapping the data positions onto an index (C), and using the connector track (B) to connect the physical genomic position with its indexed equivalent.Here methylation values (A) for 7 tissues are summarized using stacked histograms (C), whose bins represent statistics for remapped methylation probe positions.

Zimmer, C. (2008). Now: The Rest of the Genome. New York Times. Figure by M Krzywinski.

`sessions/1/11`

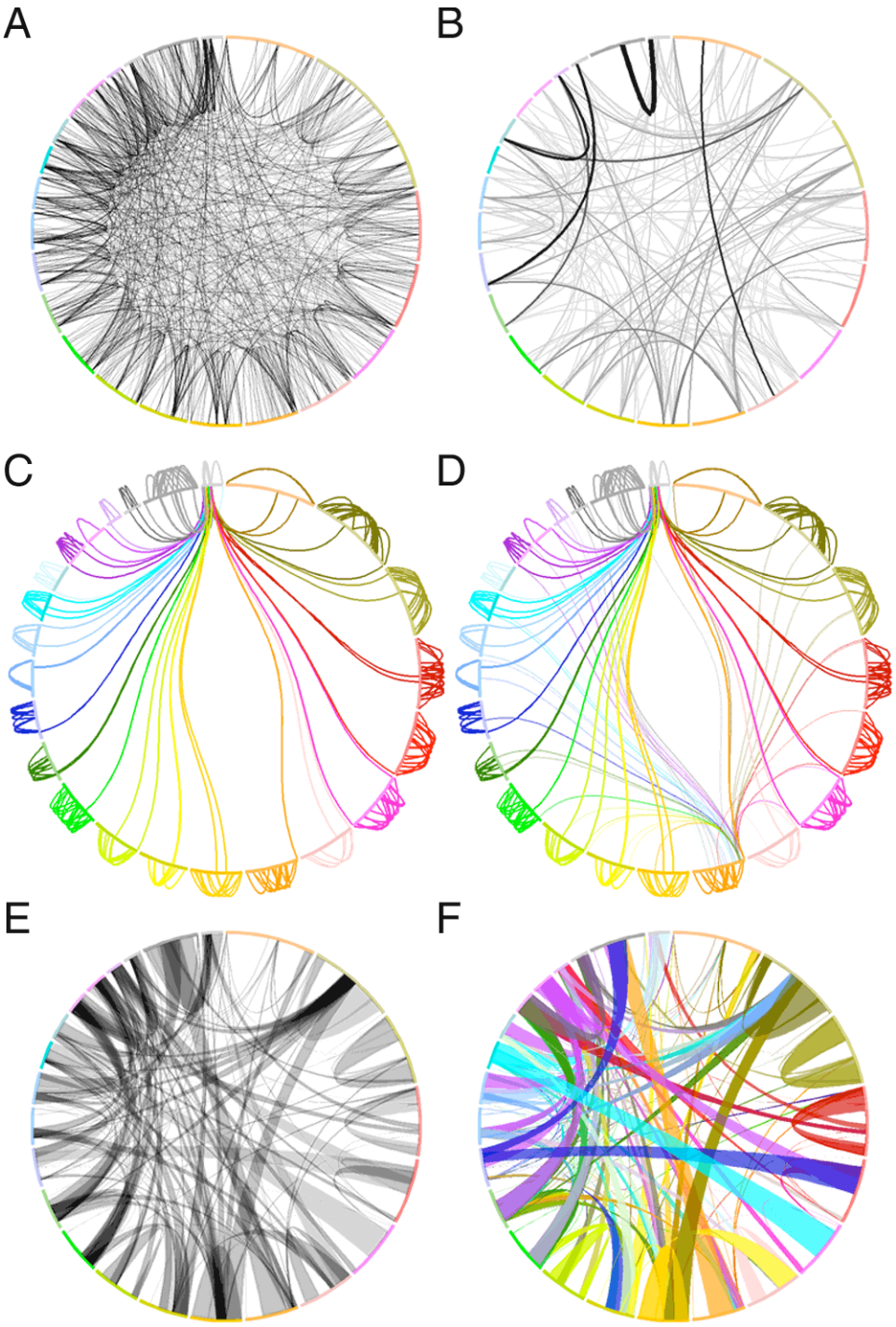


FIGURE 17

The same data set is shown in all panels. (A) Each link represents one of a subset of 2,500 segmental duplications within the human genome. (B,C,D) Rules are used to change the format of the links, by selectively adjusting color, geometry and visibility of links based on size and position. (E,F) Adjacent links are bundled into thicker links (bundles) to reduce the complexity of the figure.

`session/1/12`

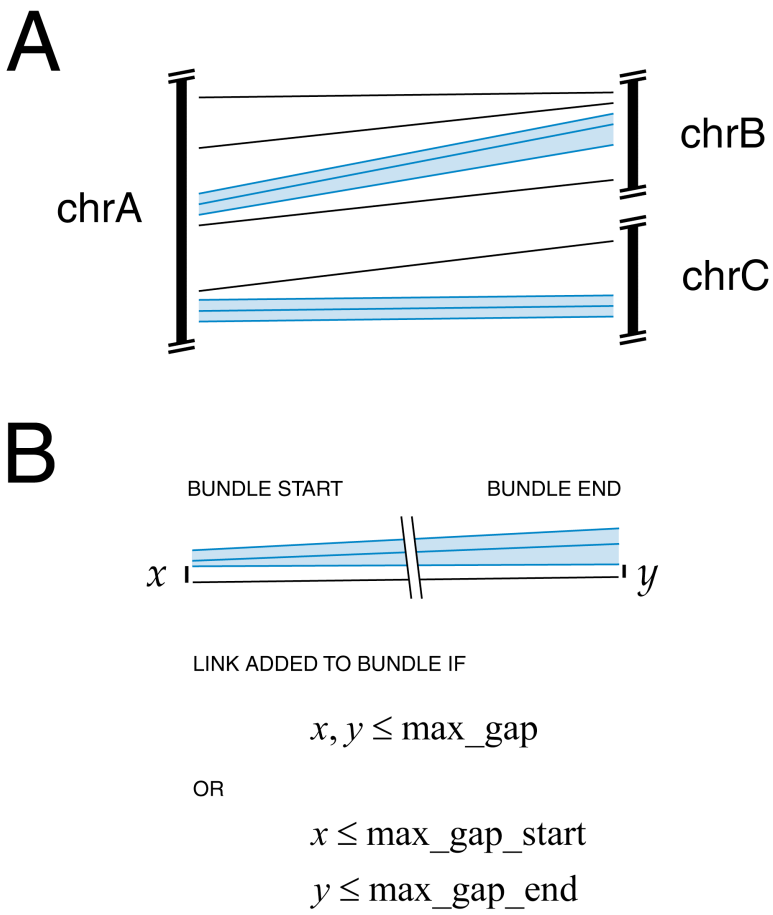


FIGURE 18

The `bundlelinks` tools is used to logically group adjacent links together, forming larger links. Links are bundled based on their size and distance to each other. Bundles are ideally drawn as ribbons, rather than lines, because bundle ends typically span a significant section of an ideogram.

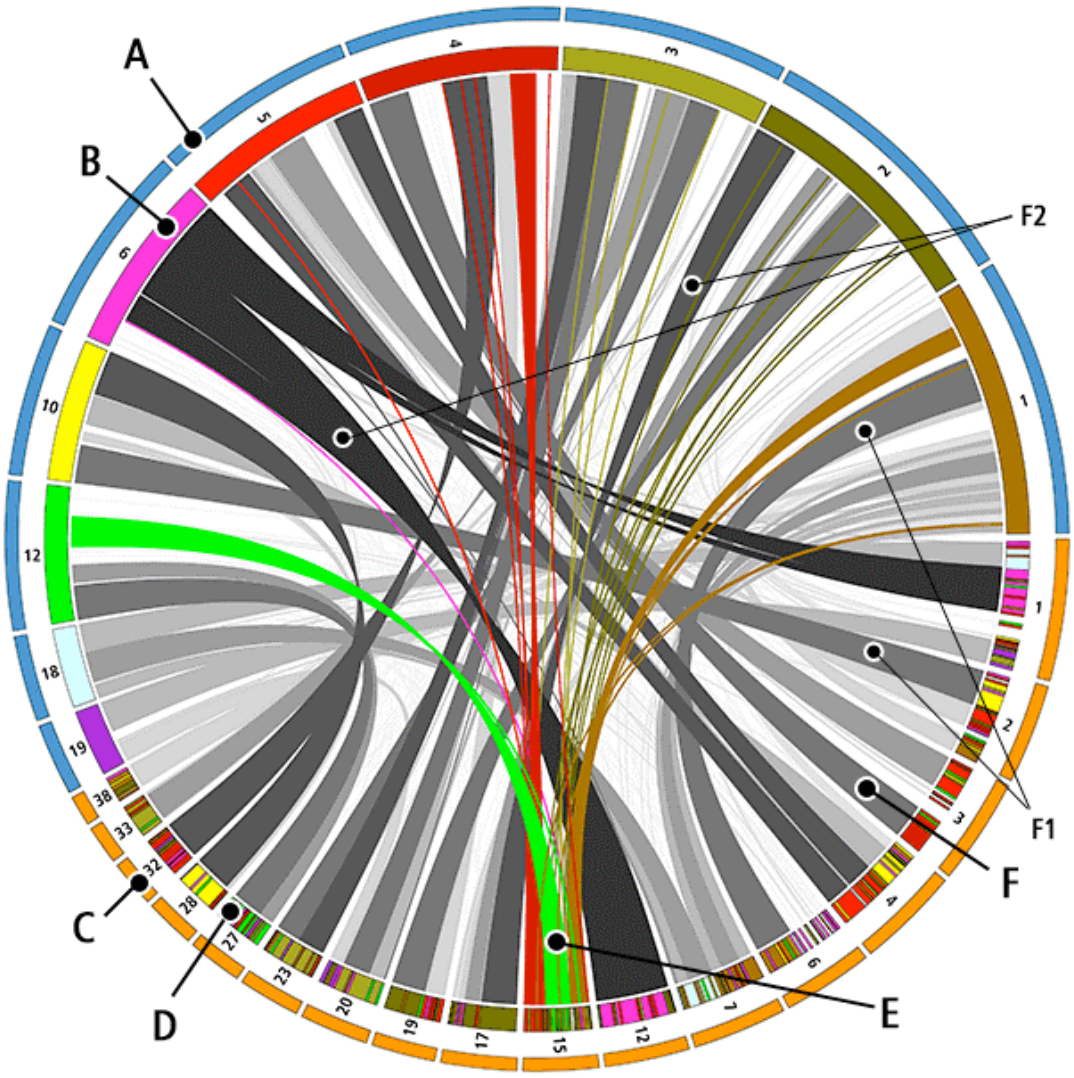


FIGURE 19

Regions of similarity between human (top, blue [A]) and dog (bottom, orange [C]) chromosomes. One dimensional similarity mapping between human [B] and dog [D] chromosomes. This mapping provides the chromosome color coding associated with grey ribbons [F]. These grey ribbons are composed of binned homology regions that fall in the same bundle (see above). The level of grey is proportional to the size of the homologous regions. Homology on chromosome 15 is highlighted with colored ribbons [E]. Ribbons that twist such as [F2] indicate inversions, whereas those that don't [F1] indicate regions of homology on the same strand.

Ostrander, E. A. (2007). "Genetics and the Shape of Dogs." American Scientist **95**(5): 406-413. Cover figure by M Krzywinski.

`sessions/1/13`

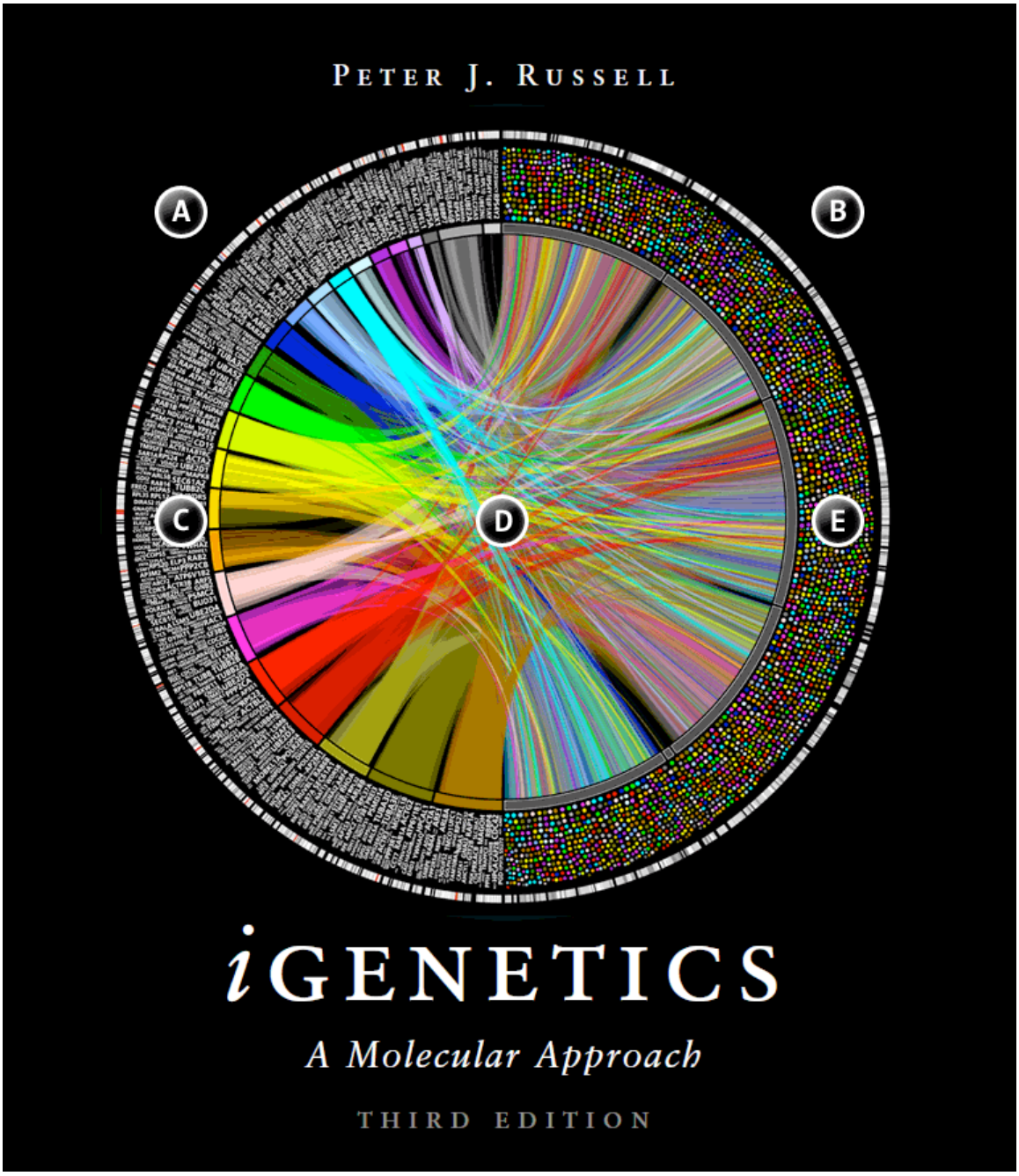


FIGURE 20

(A) Human chromosomes start at 6 o'clock and continue clockwise to 12 o'clock (1, 2, 3,. . . , 22, X, Y). (B) The order of fly chromosomes on the right is also clockwise (2L, 2R, 3L, 3R, 4, X). (C) The names of human genes that have functional equivalents in the fly. The equivalence relationship is established by degree of protein similarity in human and fly. The size of the gene name in the illustration is proportional to this similarity. (D) Links connect orthologous genes, color coded by the human chromosome. (E) The human gene's functional equivalent in fly is represented by a dot. The dot is placed at the position of the gene and its size is proportional to protein similarity.

Russell, P. J. (2010). iGenetics: A Molecular Approach, Benjamin Cummings. Cover figure by M Krzywinski.

`sessions/1/14`

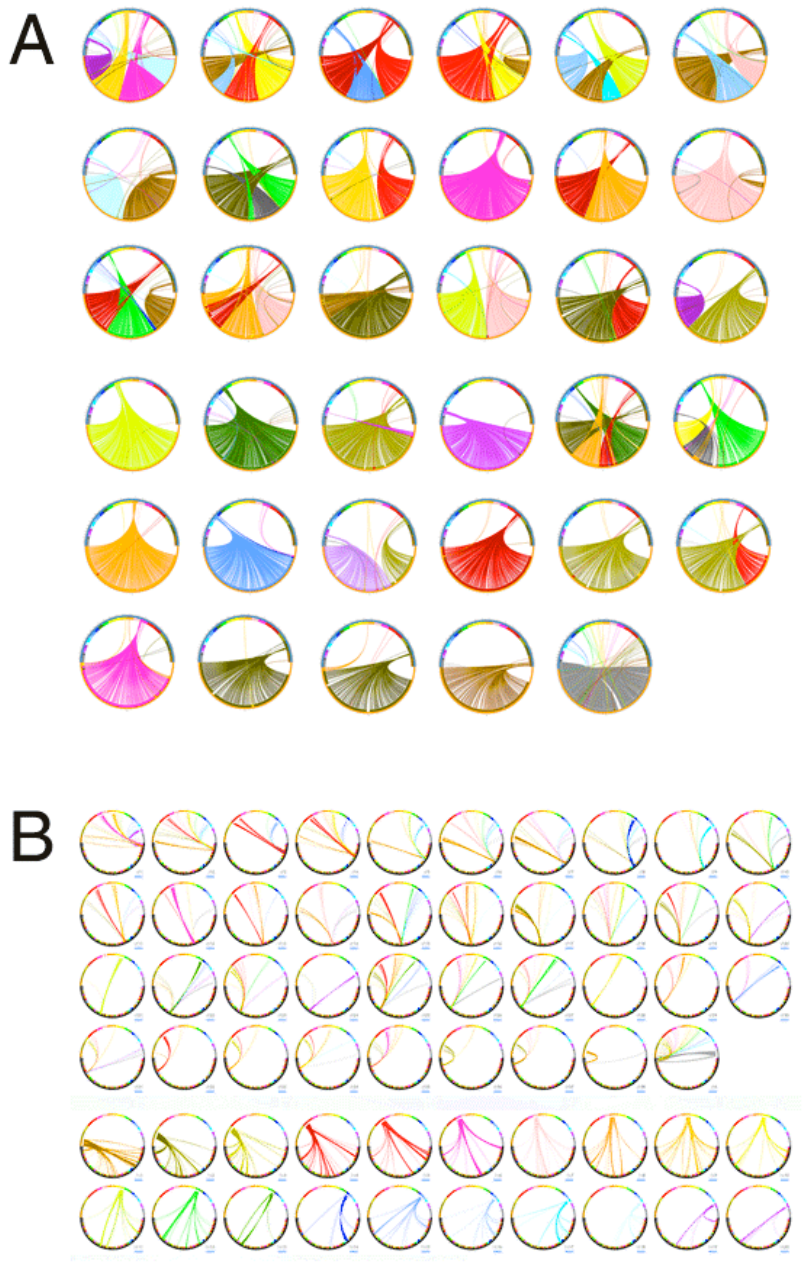


FIGURE 21

(A) Synteny between dog and human genomes. Each image represents the comparison of a single dog chromosome (bottom half of circle) with the entire human genome. The links represent similarity and are color coded by human chromosomes.

Krzywinski, M., J. Schein, et al. (2009). "Circos: an information aesthetic for comparative genomics." Genome Res **19**(9): 1639-1645.

(B) Each image contains the entire dog and human genomes (bottom and top half of circle, respectively). Links shown are based on the same data as in (A), but limited to a single chromosome (dog or human) for each image in the panel. For more information, see *mkweb.bcgsc.ca/circos/presentations/articles/amsci_cover*

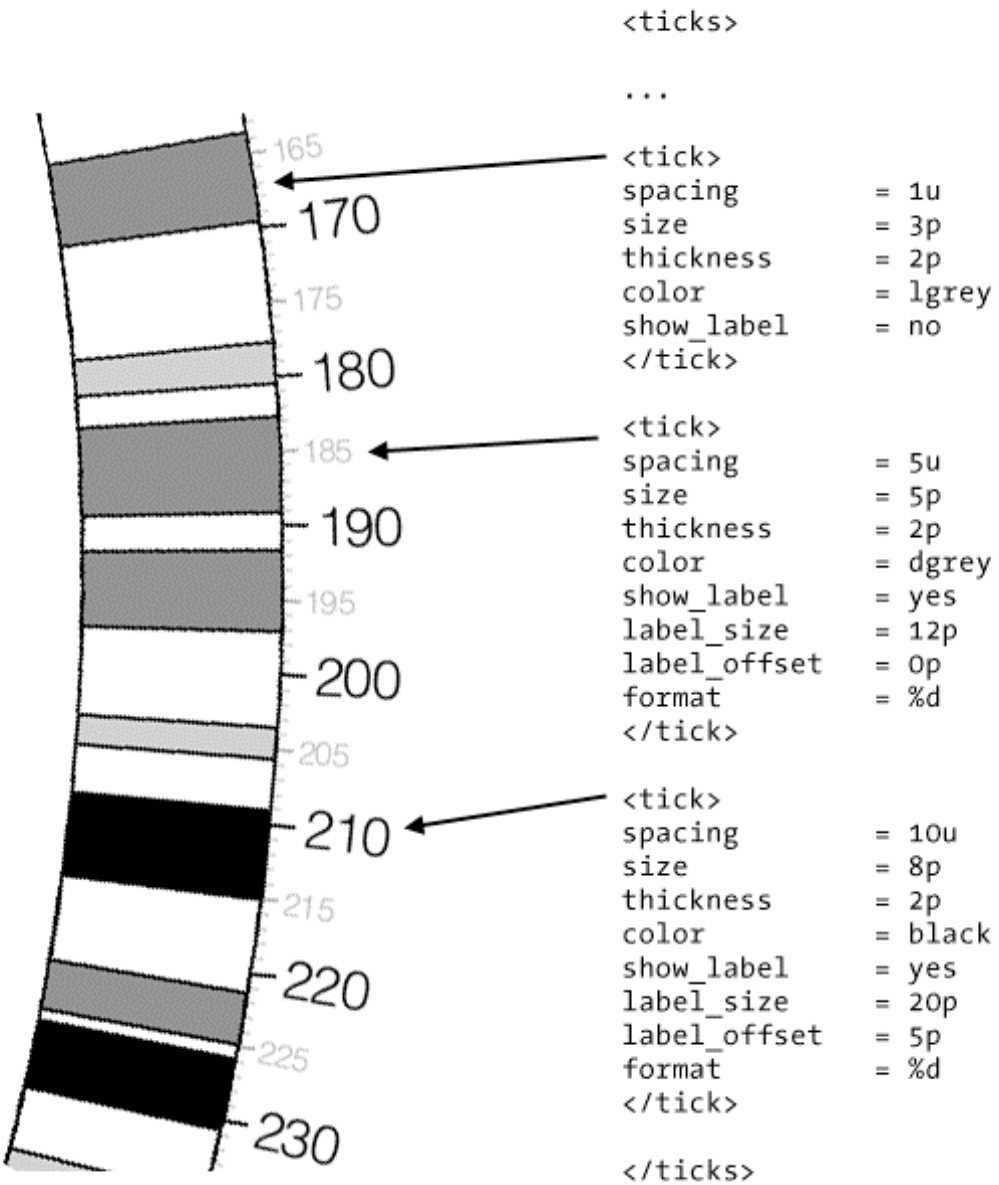


FIGURE 22

Ticks are divided into groups. Here three groups are shown, with ticks spaced every 10Mb, 5Mb and 1Mb. Each group is defined independently. Notice that the 1Mb group has no labels, 5Mb group has a small label and 10Mb group has a larger label with an offset.

`sessions/1/15`

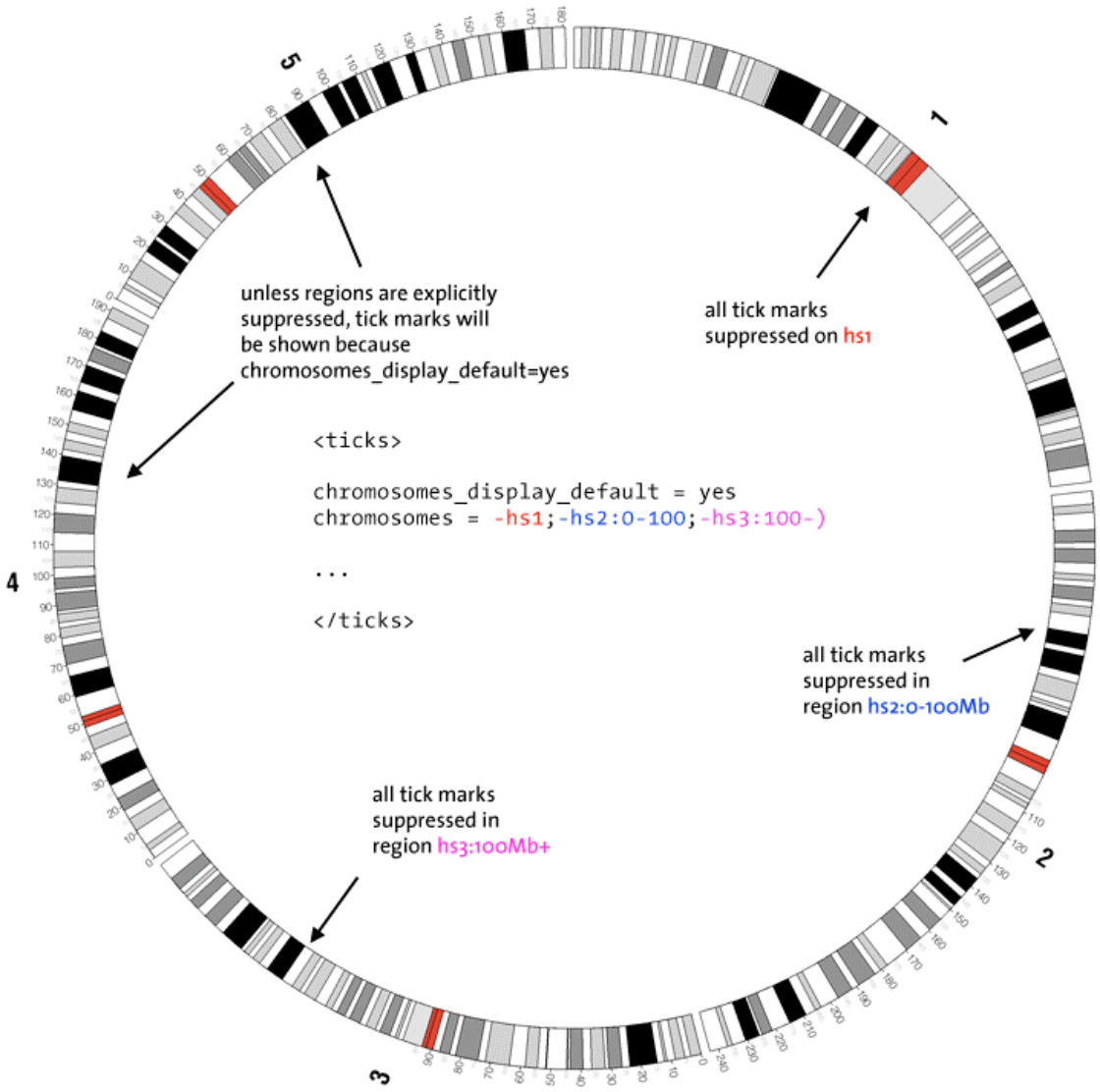


FIGURE 23

All ticks, as well as each group, can be suppressed on chromosomes or within regions.Tick display can be turned off by default, with the `chromosomes` parameter specifying where ticks should be drawn. Alternatively, display can be turned on, with the parameter specifying where ticks should not appear.

`sessions/1/15`

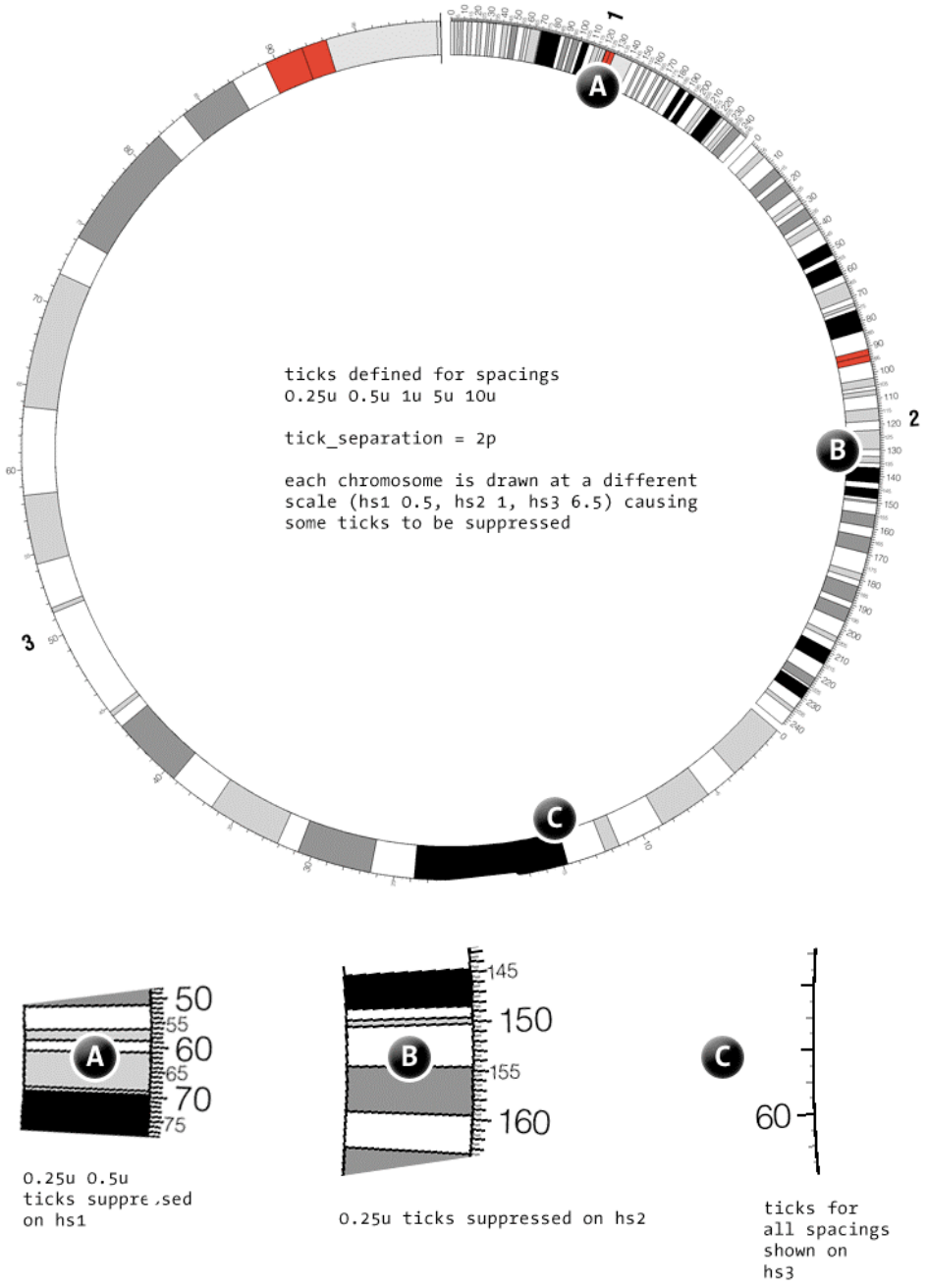


FIGURE 24

Chromosomes 1, 2 and 3 are shown at different scales (*0.5x*, *1x* and *6.5x*). Ticks are defined for spacings of 0.25, 0.5, 1, 5 and 10mb.Minimum tick separation is defined to be 2 pixels. On chromosome 1 (A), ticks spaced at 0.25Mb and 0.5Mb are automatically hidden, because they would be drawn closer than minimum separation. On chromosome 2 (B), where the scale is larger, the 0.5Mb ticks are drawn, but the 0.25Mb ticks are still suppresed. Chromosome 3, which is significantly stretched, can accommodate all ticks.

`sessions/1/16`

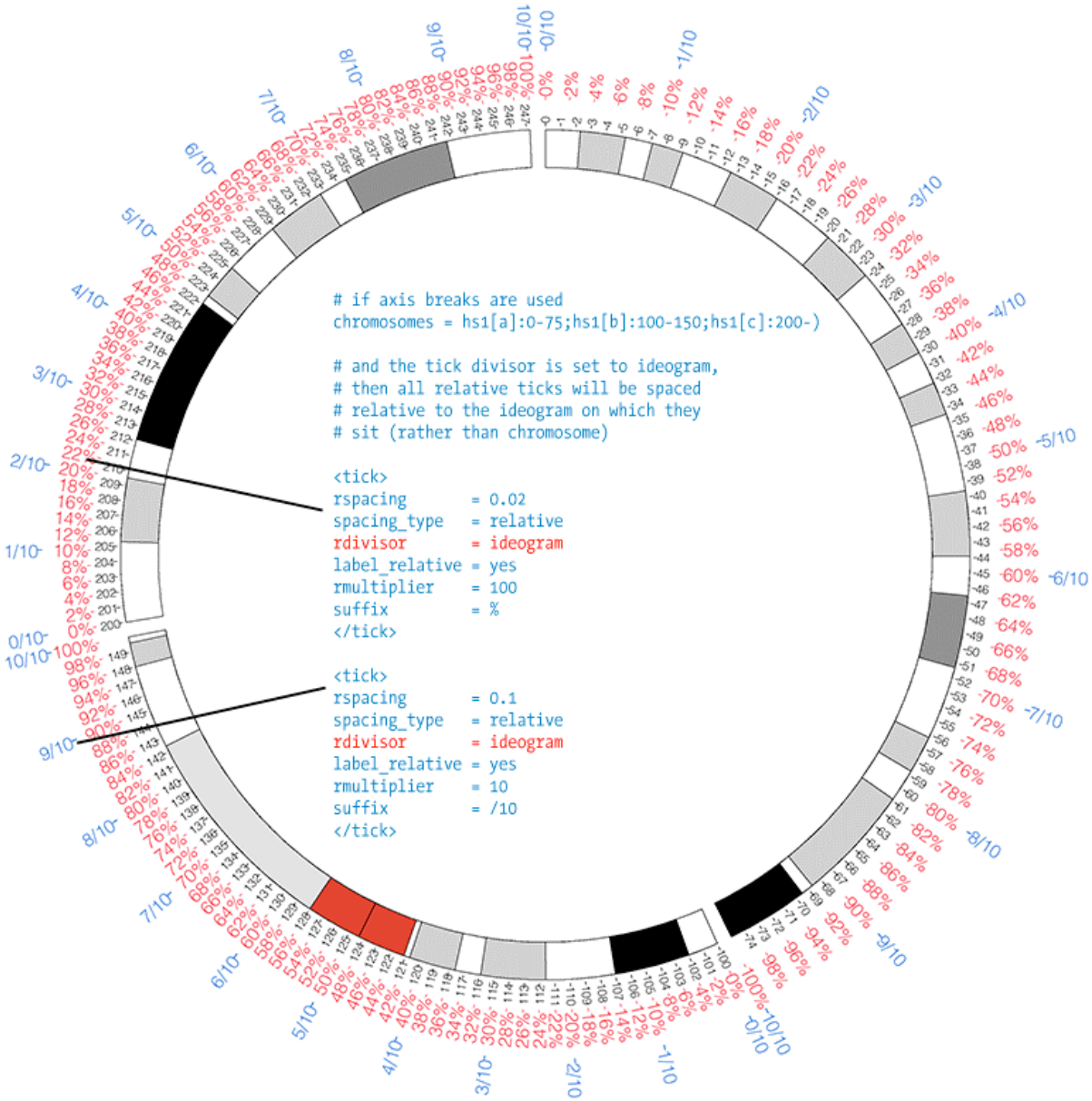


FIGURE 25

Absolute (black) and relative (red, blue) ticks can be combined. Ticks can be given any suffix. Here, the red tick track has a suffix `%` and the blue track has a suffix `/10`. Note the rdivisor parameter, which alters the offset of relative tick marks to be relative to the ideogram, not the chromosome. This makes a difference if the ideogram is showing only a region of a chromosome. Note that the first relative tick on each ideogram shows as 0%, whereas the first absolute tick shows the start of the ideogram on the chromosome (0, 100, 200).

`sessions/1/17`

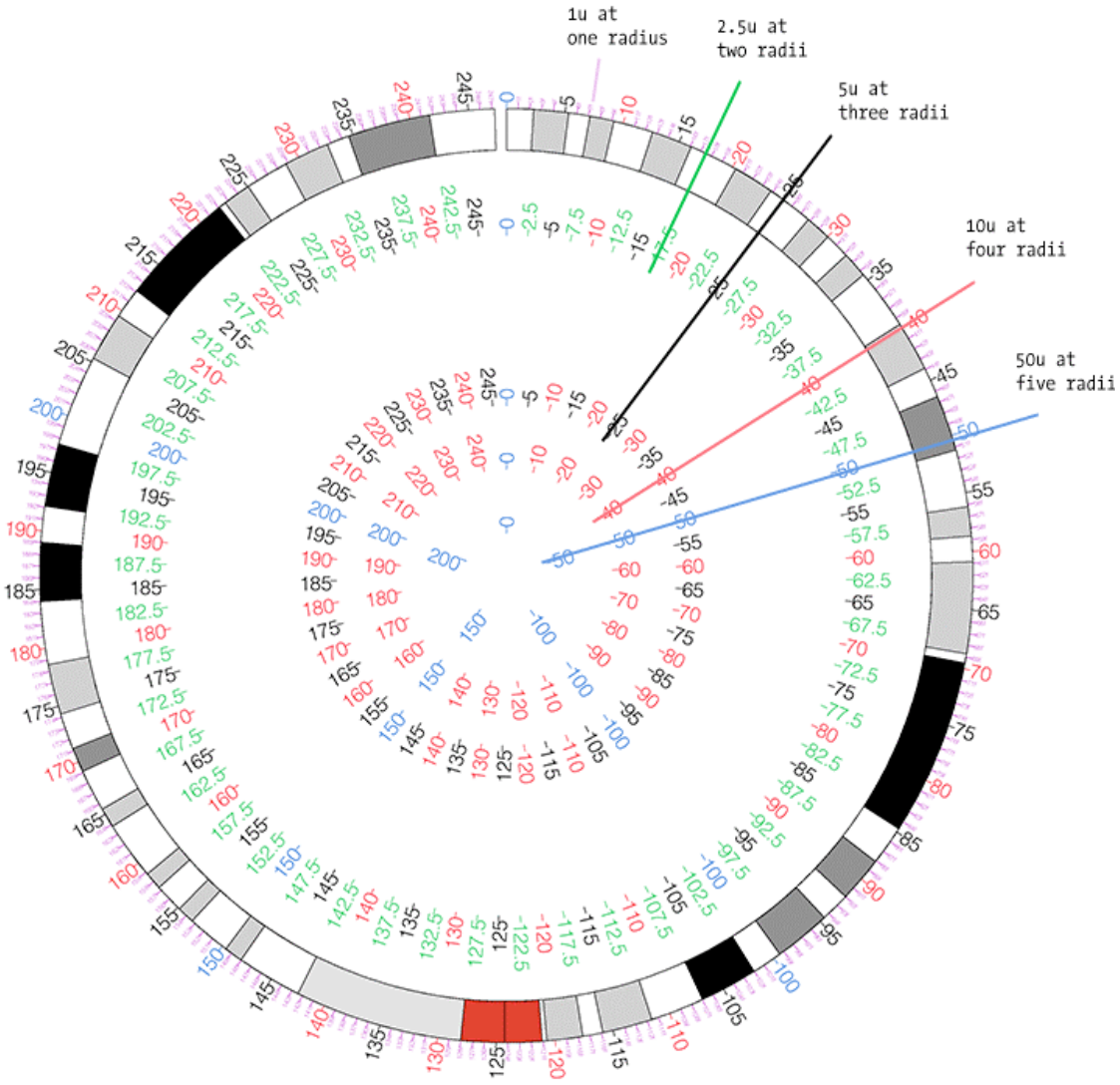


FIGURE 26

Five tick groups are drawn at five different radial positions. Each tick group appears in one or more rings. The outer tick ring contains all ticks, and the inner ring only the ticks spaced at 50Mb. Limiting display of tick groups in this way helps maintain a more uniform tick density at all radial positions.

`sessions/1/18`

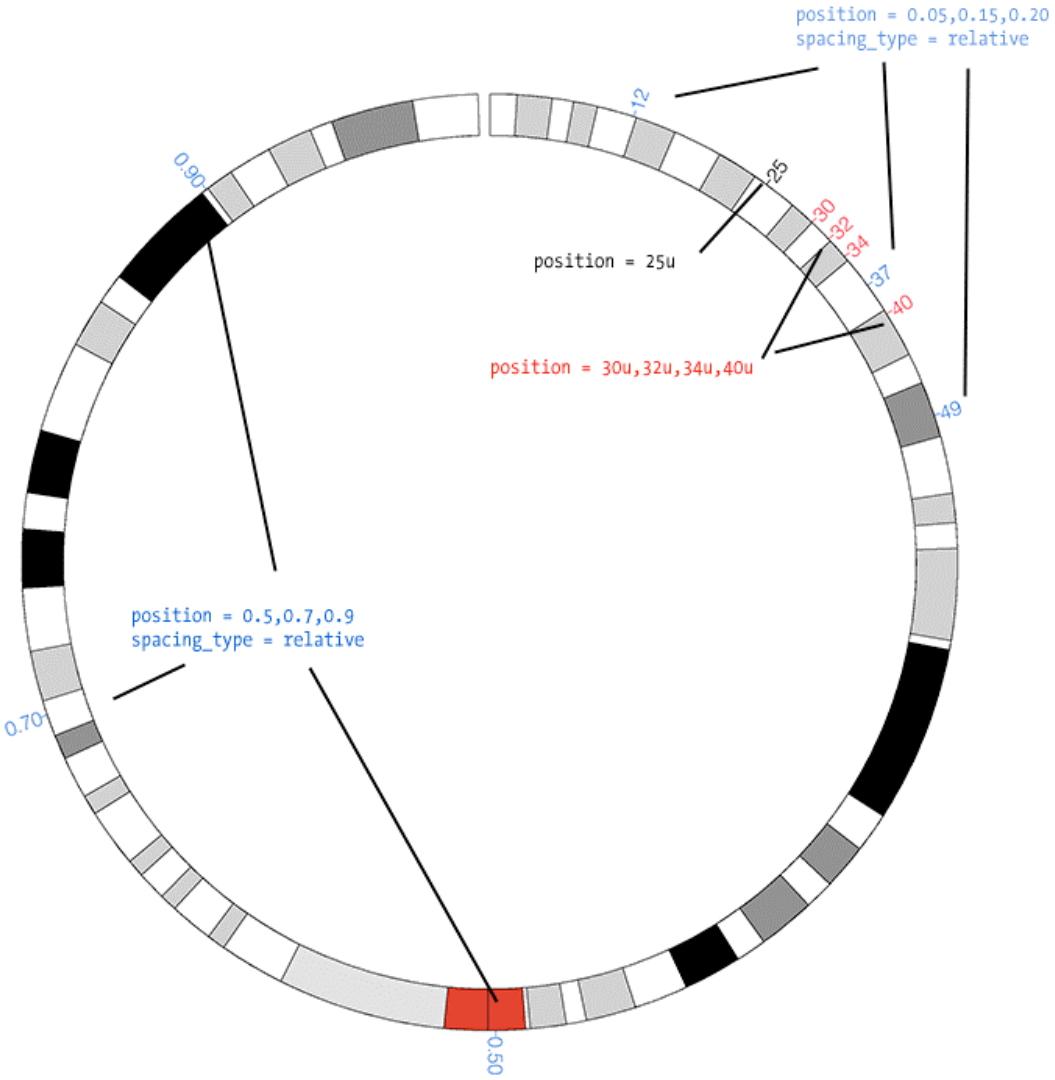


FIGURE 27

Ticks in a group, instead of having uniform spacing, can be can be placed at arbitrary locations. Positions can be either relative or absolute.

`sessions/1/19`

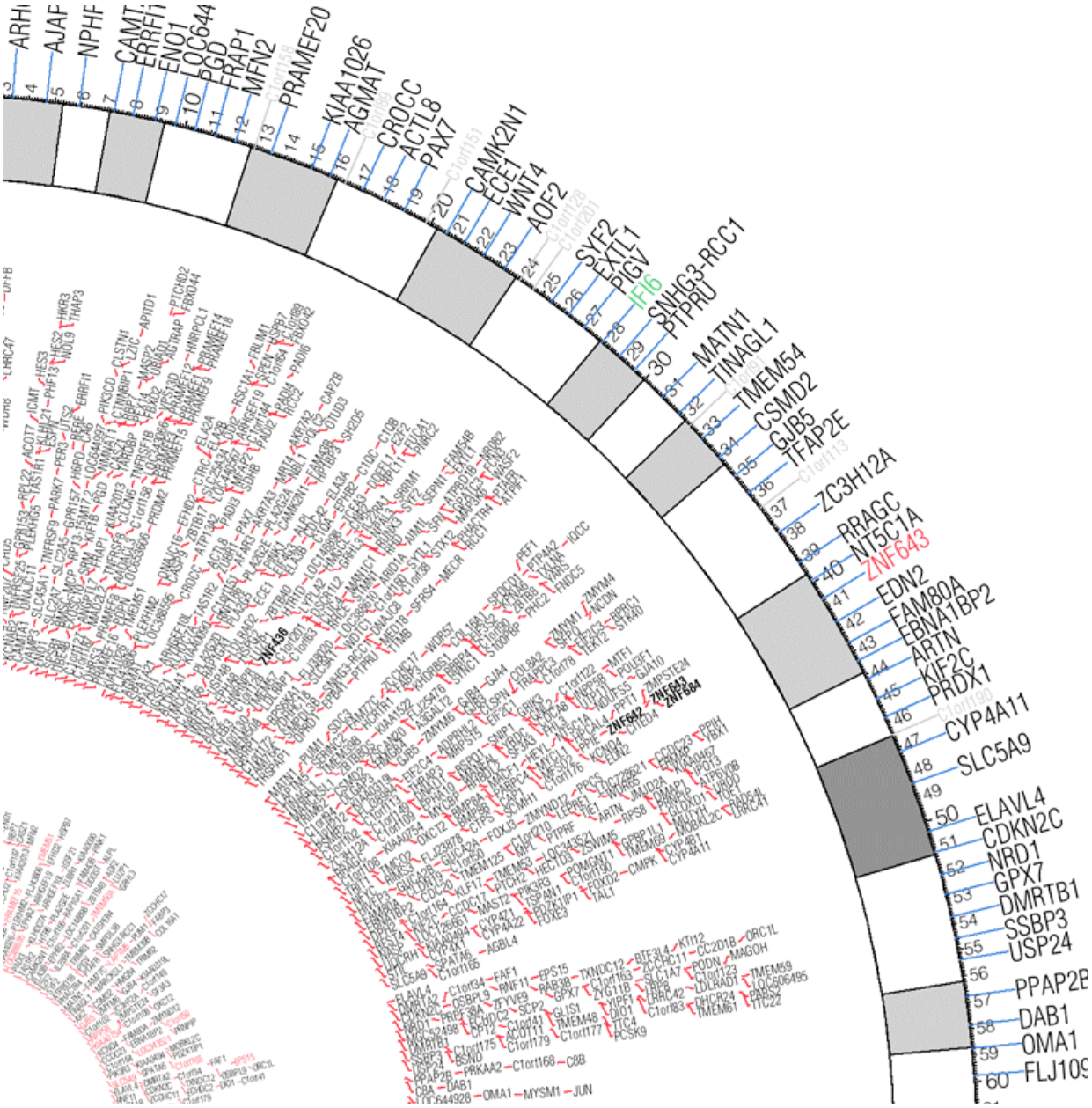


FIGURE 28

Labels in text tracks stack automatically within the track area to avoid overlap. Lines can be used to relate each label to its position, a helpful feature when labels are locally rearranged for layout.

`sessions/1/21`

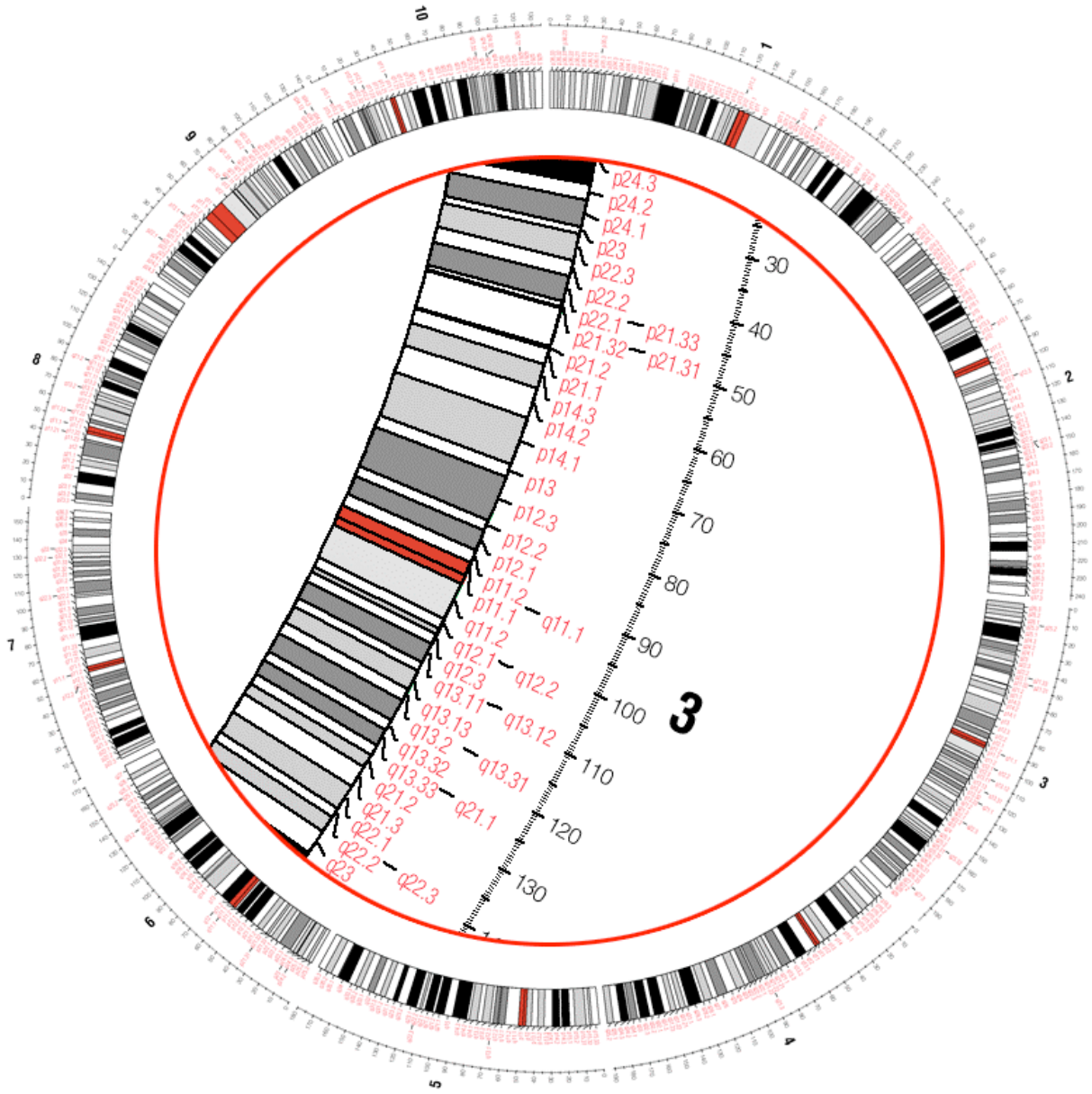


FIGURE 29

In a text track, neigbouring labels are stacked to avoid overlap. A portion of the figure is shown in a zoomed inset, inside the ideogram circle. This inset was created during post-processing and is not a feature of Circos.

SESSIONS/1/20

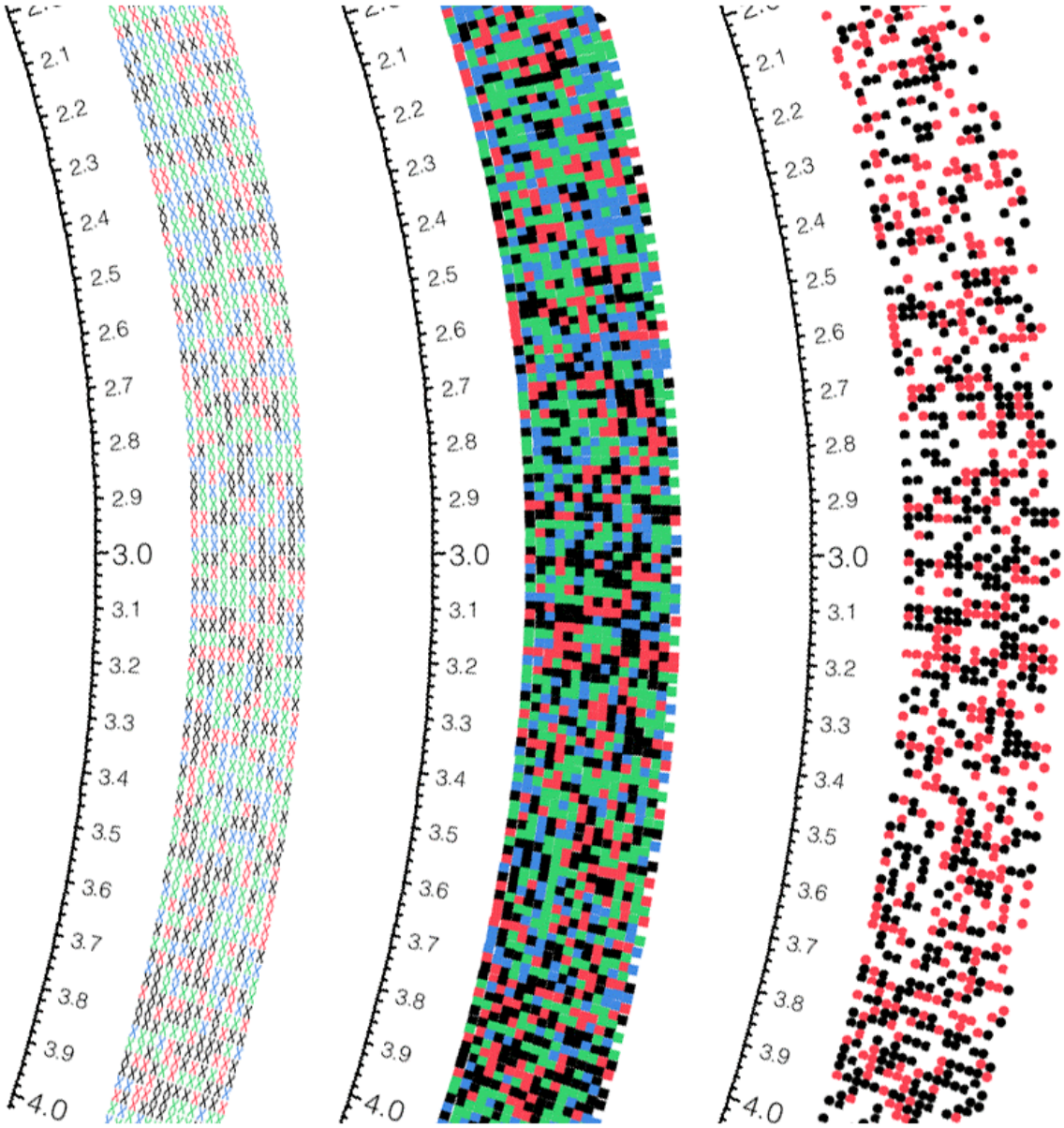


FIGURE 30

Three tracks showing sequence data. Each label corresponds to a base, colored by the identity of the base. In the first track, each base label is changed to X using rules. In the second track, a wingding symbol font is used, and the label is changed to ASCII 110 (*n*), which corresponds to a square glyph in this font. In the third track, the label is changed to ASCII 108 (*l*), which is a circle.

`sessions/1/22, images 1,2,3`

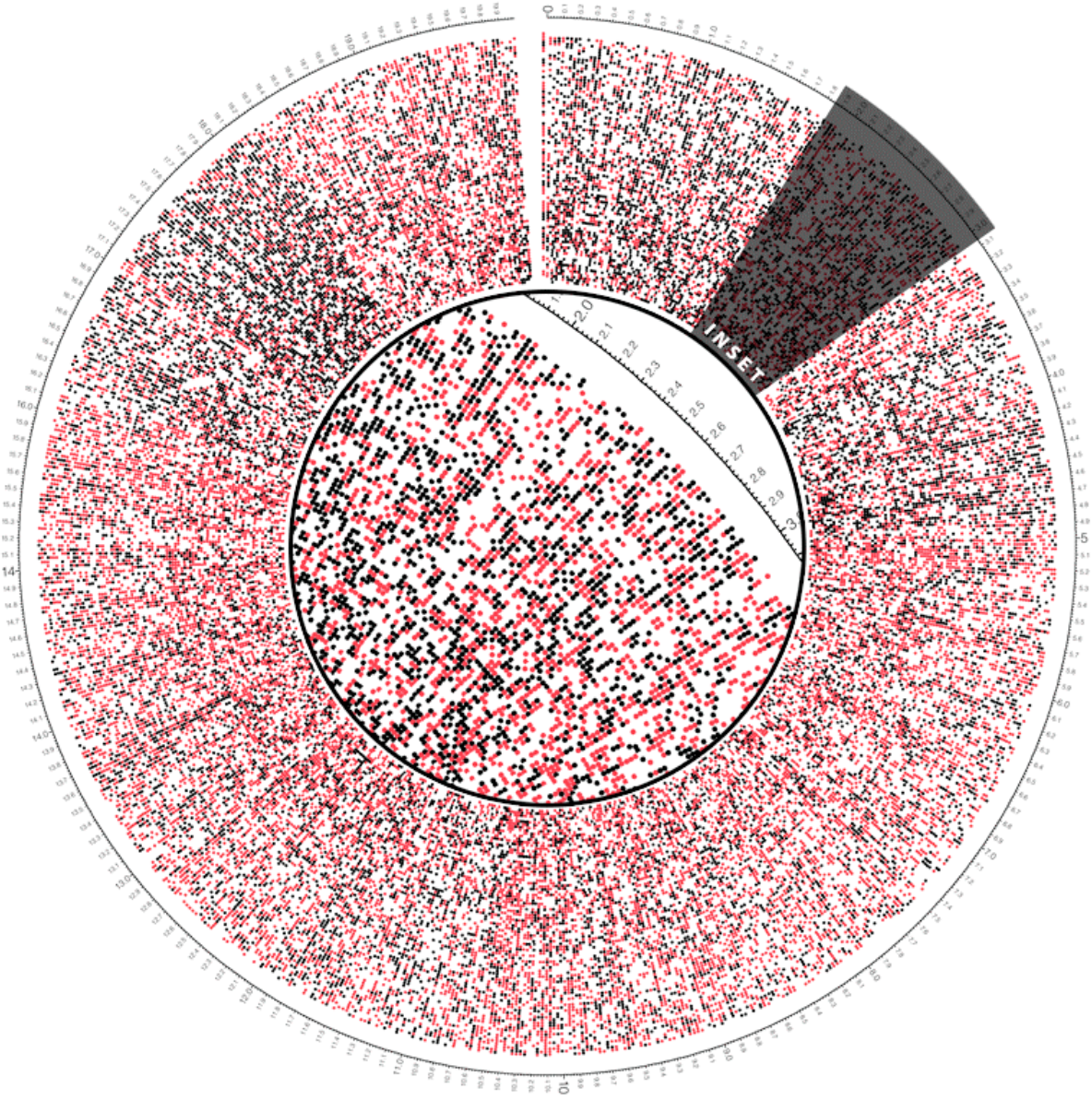


FIGURE 31

A glyph track filling the entire image.

`sessions/1/22, image 4`

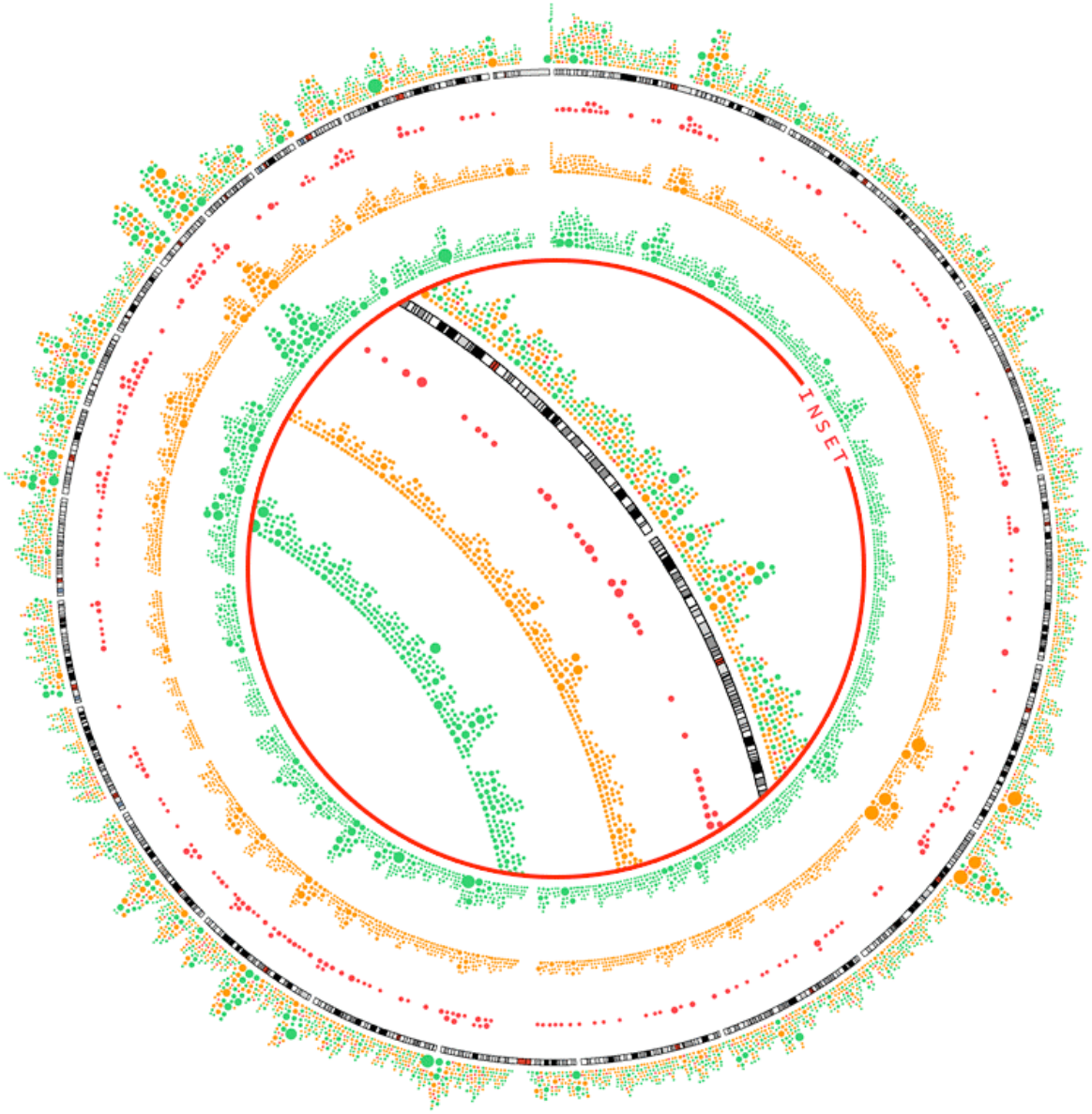


FIGURE 32

A single gene density data file,is used to populate four tracks. Individual density data points are categorized based on the gene category they correspond to. Rules are used to show specific categories in a track and to change the label from the category name (e.g. *cancer*) to an *l*, which is a circle in the wingding font.

`sessions/1/22, image 5`

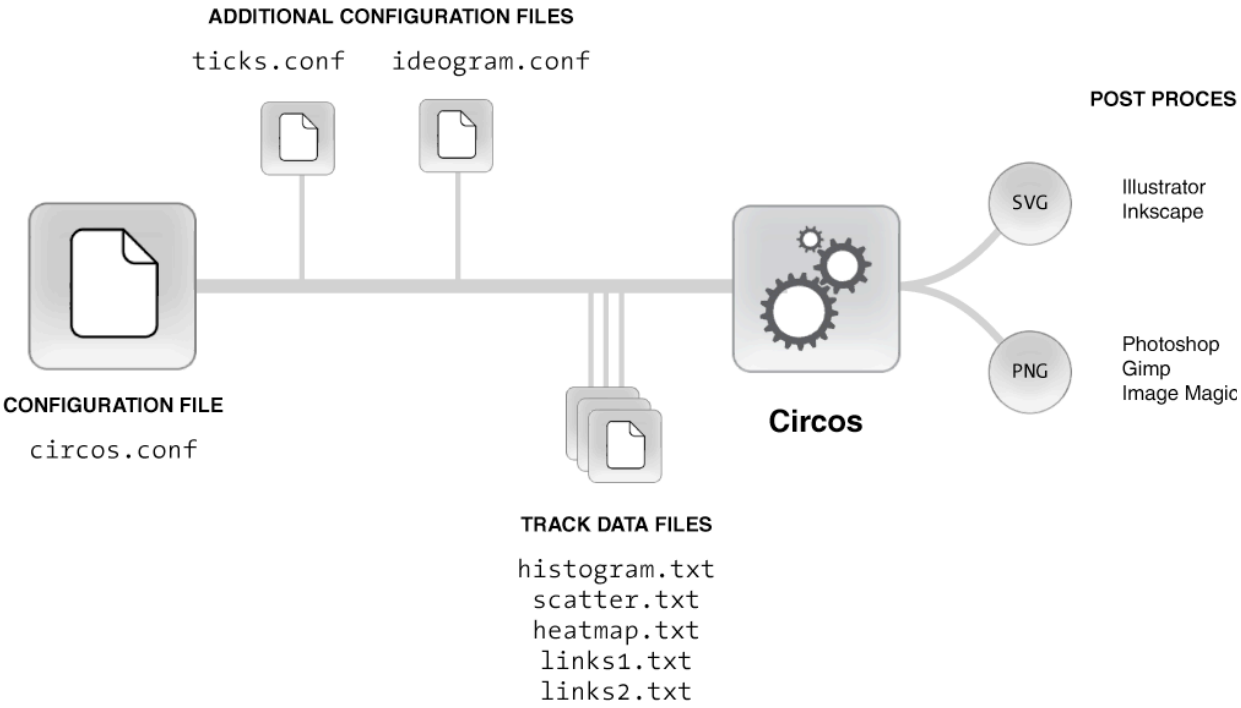


FIGURE 33

Typically a central configuration file which defines data track information (`circos.conf`) imports other configuration files that store parameters that change less frequently (tick marks, ideogram size, grid, etc).

Data for each data track is stored in a file and the same file can be used for multiple tracks.

PNG image output is ideal for immediate viewing, web-based reporting or presentation.

SVG output is most suitable for generating very high resolution line art for publication and for customizing aspects of the figure.

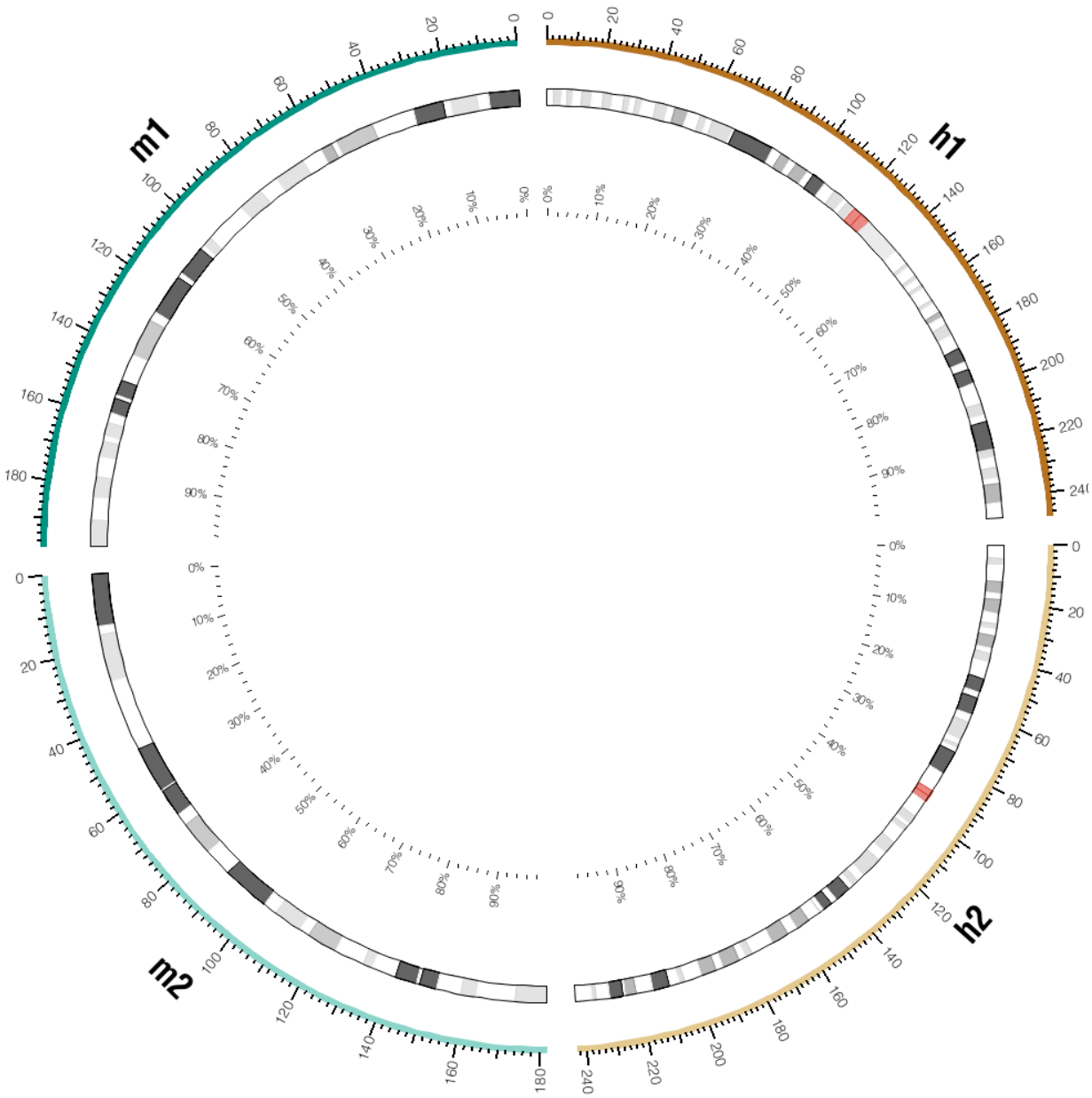


FIGURE 34

This is the image that you will construct during the first practical session.

`sessions/2/12`

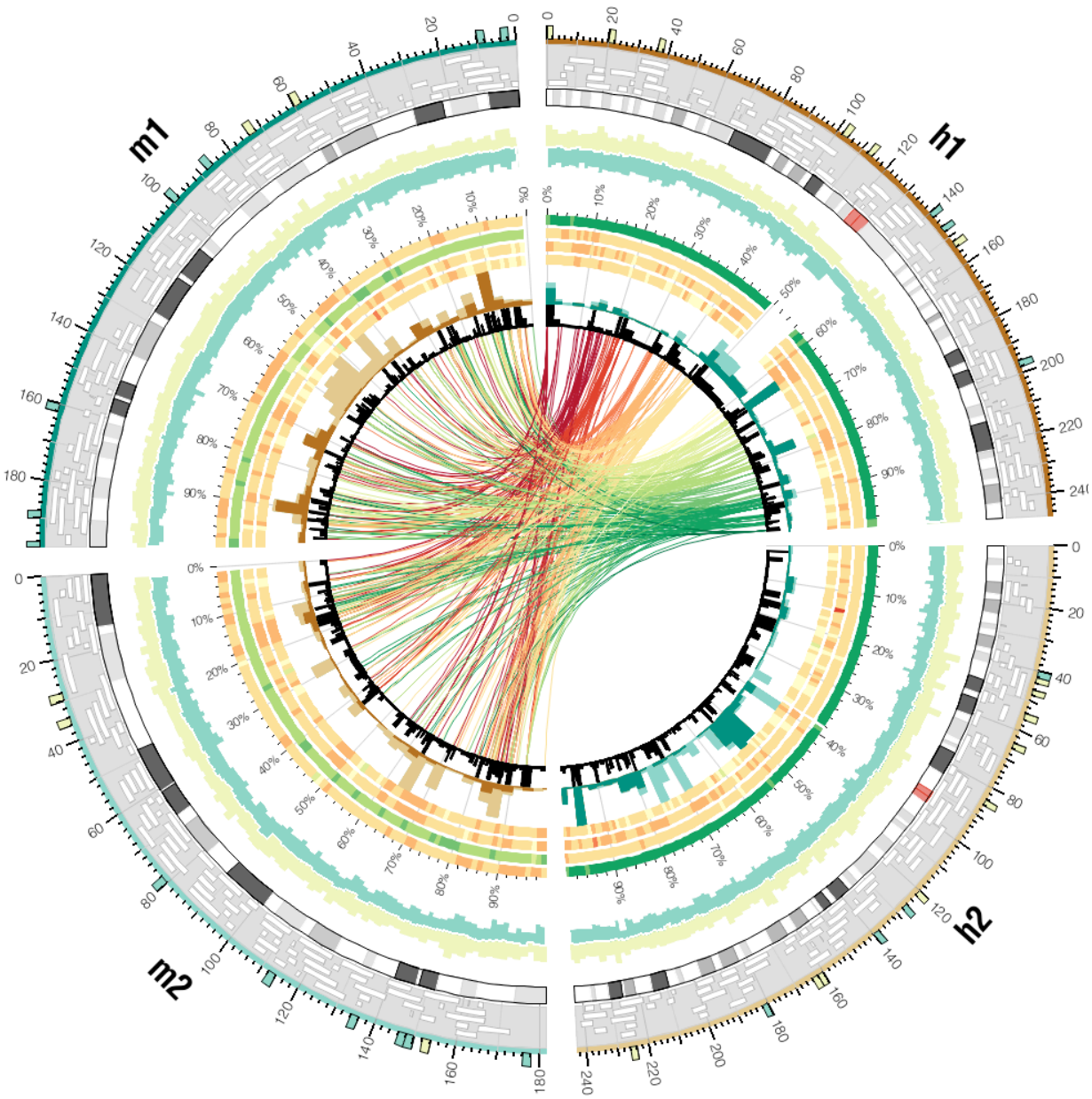


FIGURE 35

This is the image that you will construct during the second practical session.

`sessions/3/8`

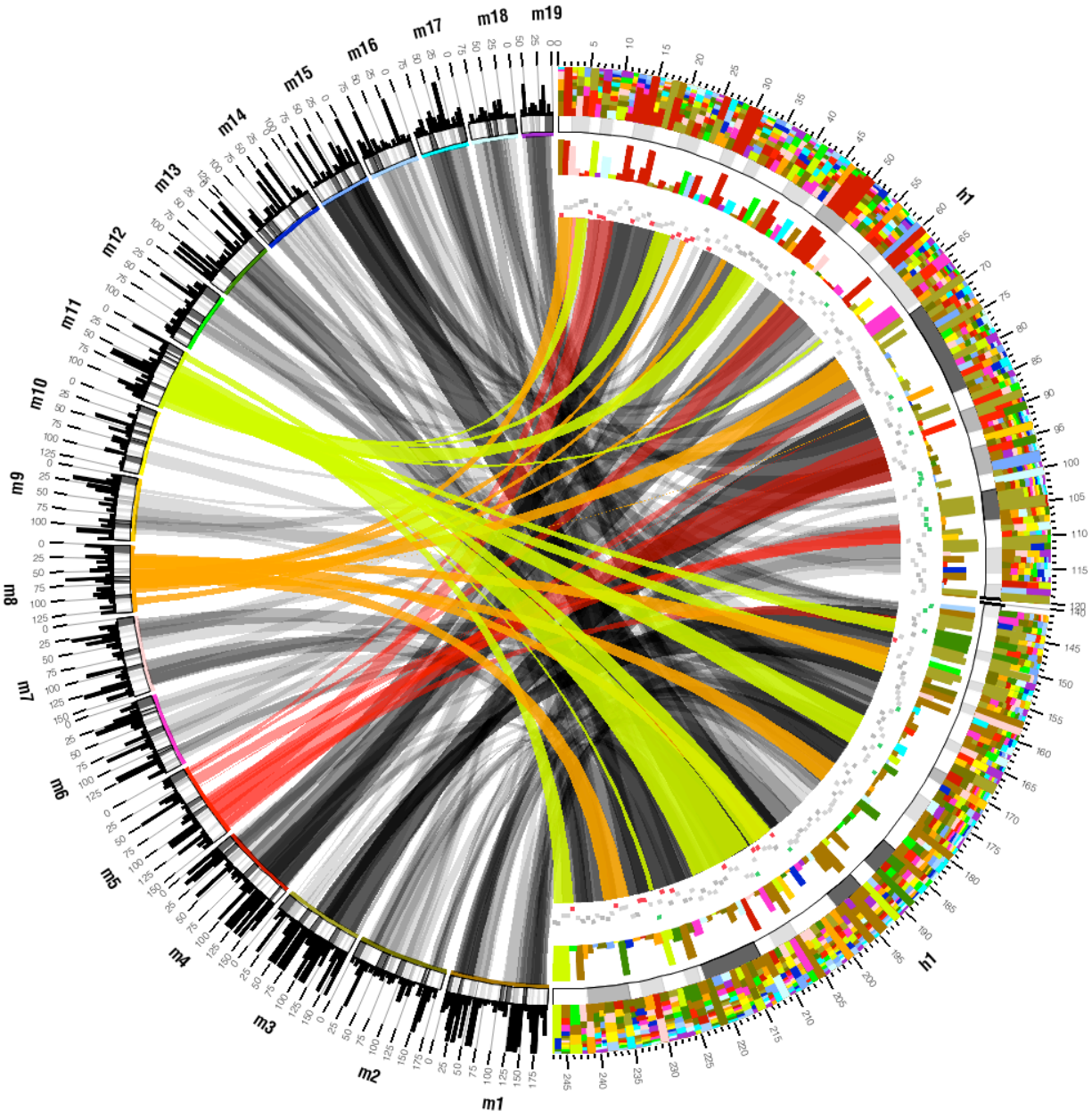


FIGURE 36

This is the image that you will construct during the third practical session.

`sessions/4/8`

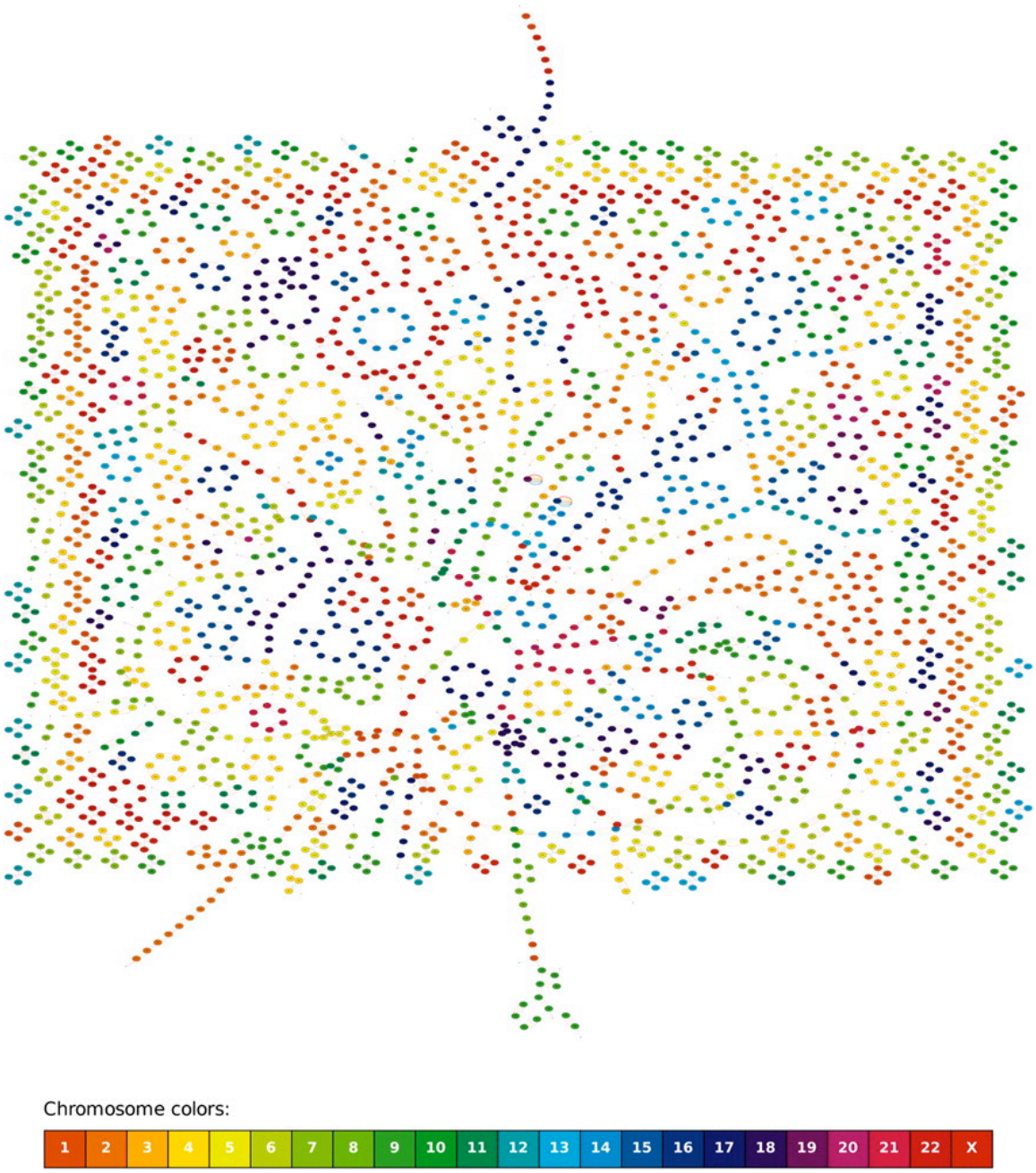


FIGURE 37

The breakspoint graph *G*(*M*,*R*,*D*,*Q*,*H*,*C*) (obverse edges are not shown) of six mammalian genomes.

Alekseyev, M.A. and P.A. Pevzner, Breakpoint graphs and ancestral genome reconstructions. Genome Res, 2009. **19(5): p. 943-57.**

COMMENTARY

This is a very attractive figure, but it does not communicate information about the graph. The representation for this visualization is too complex for a human reader to parse.

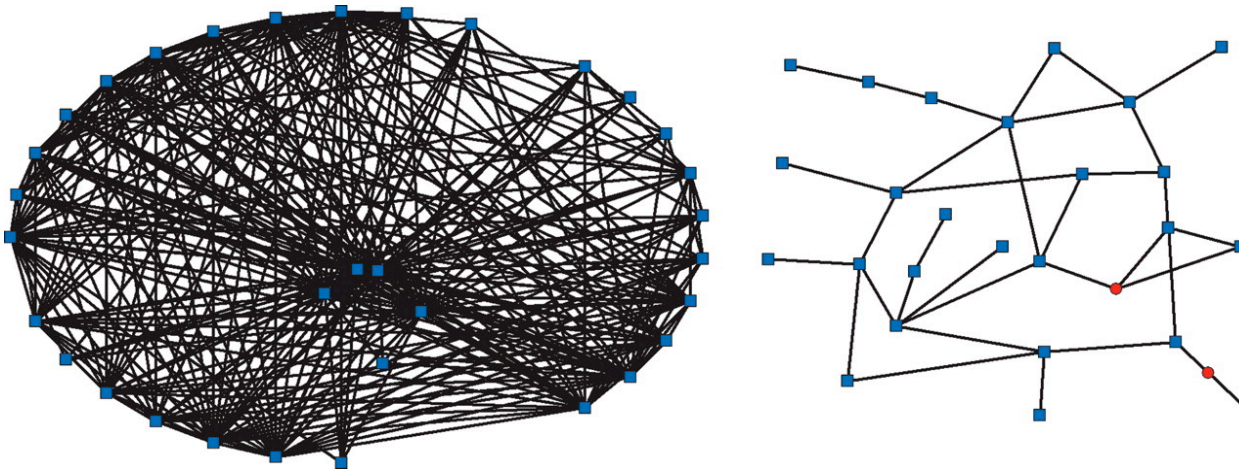


FIGURE 38

The largest E- and N-gene families in yeast.

Shakhnovich, B.E. and E.V. Koonin, Origins and impact of constraints in evolution of gene families. Genome Res, 2006. 16(12): p. 1529-36.

COMMENTARY

This figure is also too complex for a human reader to parse. The connections in the right panel are too dense and it is unclear what components of the pattern on the right are significant.

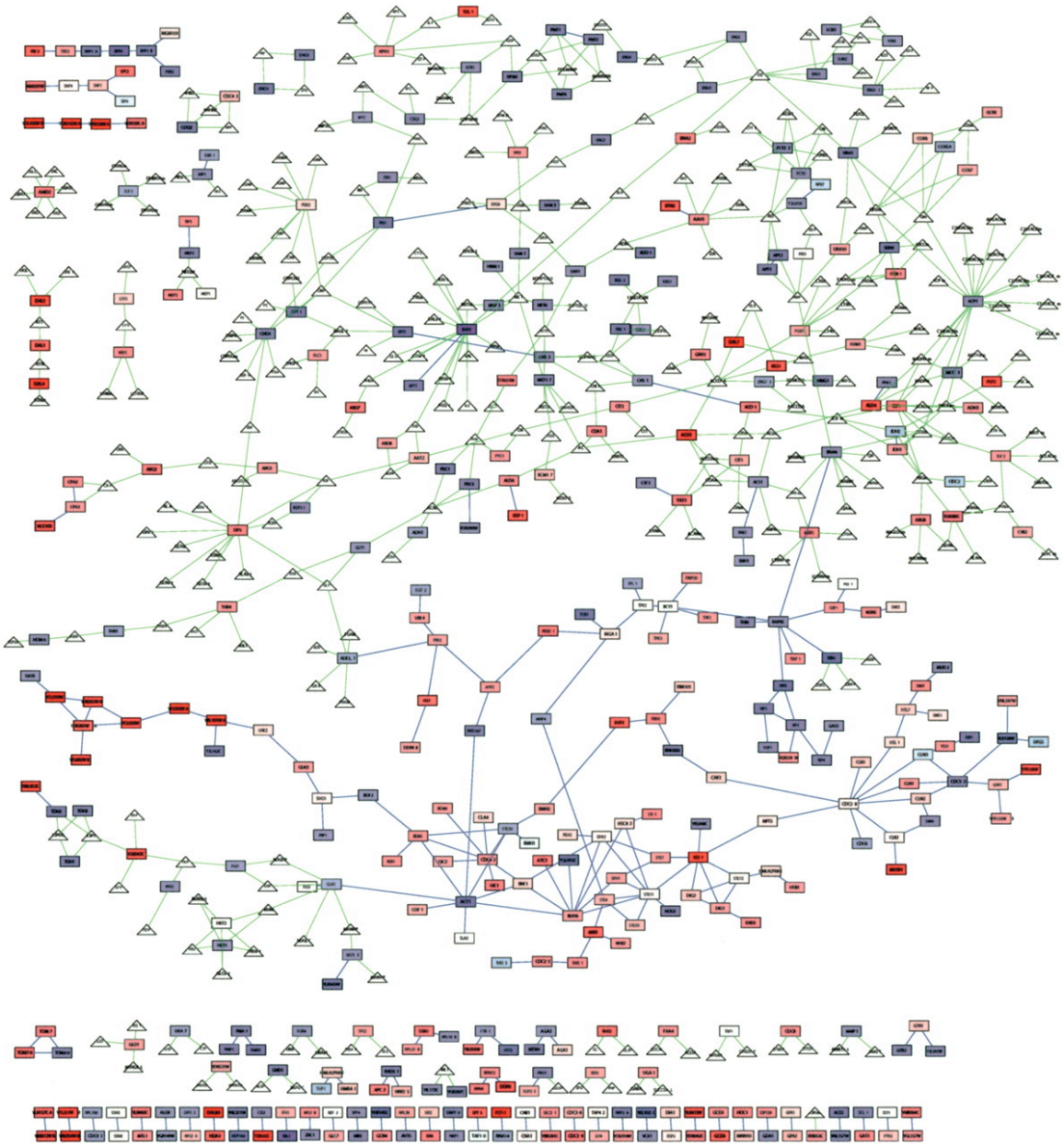


FIGURE 39

Integrated filamentation network.

Prinz, S., et al., Control of yeast filamentous-form growth by modules in an integrated molecular network. Genome Res, 2004. 14(3): p. 380-90.

COMMENTARY

This figure is too complex to parse. Its components are too small to be legible. The reader is left without an entry point into the figure and without a clear message.

FIGURE 40

The cover design depicts hairpin secondary structure of RNA sequences, precursors to microRNA.

Genome Res, 2004. 14(10a): cover.

COMMENTARY

This is a great cover design that answers the question: when should one show all the data?

The answer: Show all the data when the data present an emergent pattern that cannot be easily parametrized. Indeed, the hairpins in this image are laid out in the shape of a human form. This is immediately recognizable by a human reader, but very difficult to ascertain algorithmically.

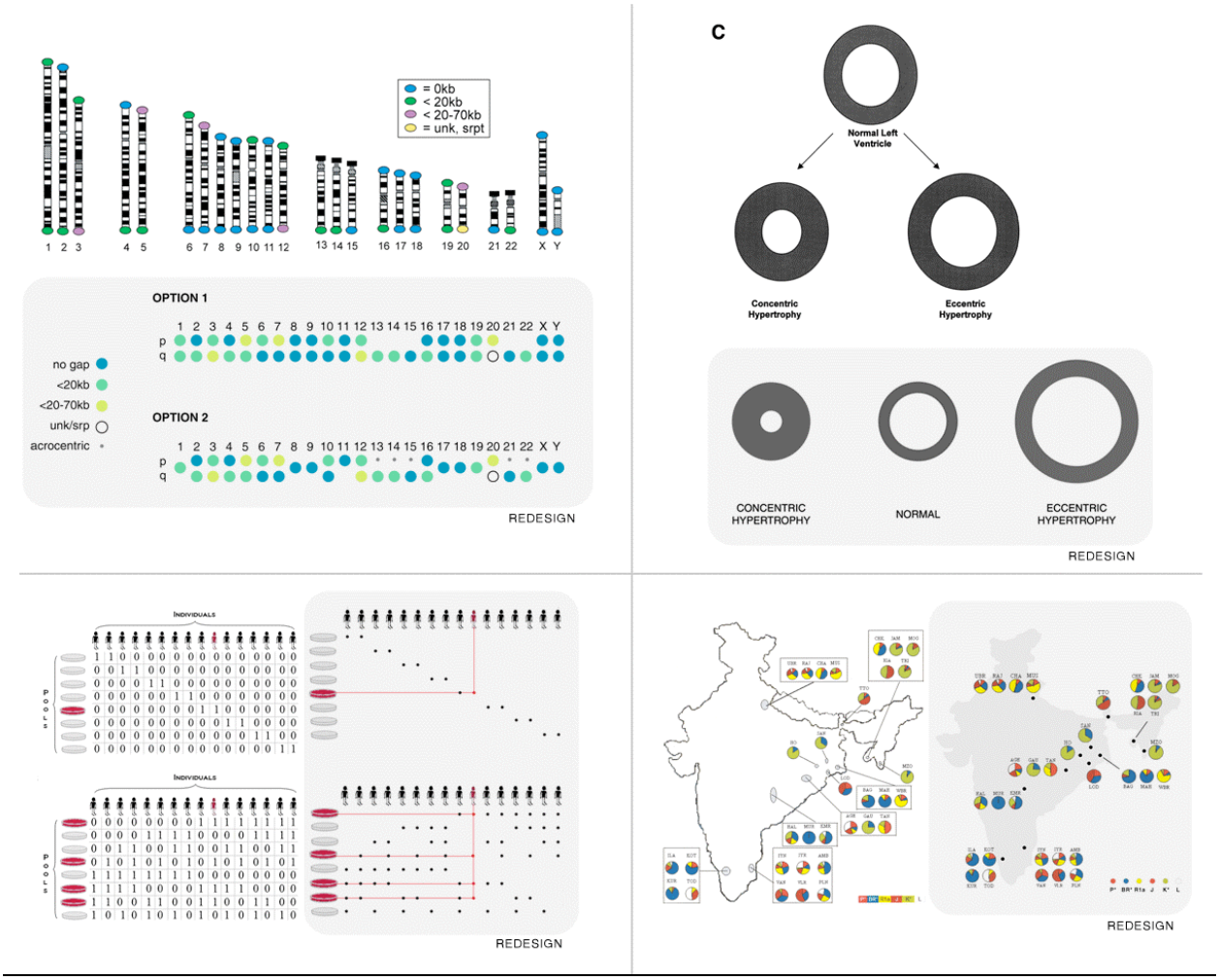


FIGURE 41

Examples from the literature will be used to illustrate common challenges in creating visualizations. A redesigned version of the figure (all redesigns by Martin Krzywinski) will be shown to illustrate how to mitigate problems such as redundancy, excess ink and poor color contrast.

Riethman, H., et al., Mapping and initial analysis of human subtelomeric sequence assemblies. Genome Res, 2004. 14(1): p. 18-28.

Nadeau, J.H., et al., Pleiotropy, homeostasis, and functional networks based on assays of cardiovascular traits in genetically randomized populations. Genome Res, 2003. 13(9): p. 2082-91.

Prabhu, S. and I. Pe'er, Overlapping pools for high-throughput targeted resequencing. Genome Res, 2009. 19(7): p. 1254-61.

Basu, A., et al., Ethnic India: a genomic view, with special reference to peopling and structure. Genome Res, 2003. 13(10): p. 2277-90.

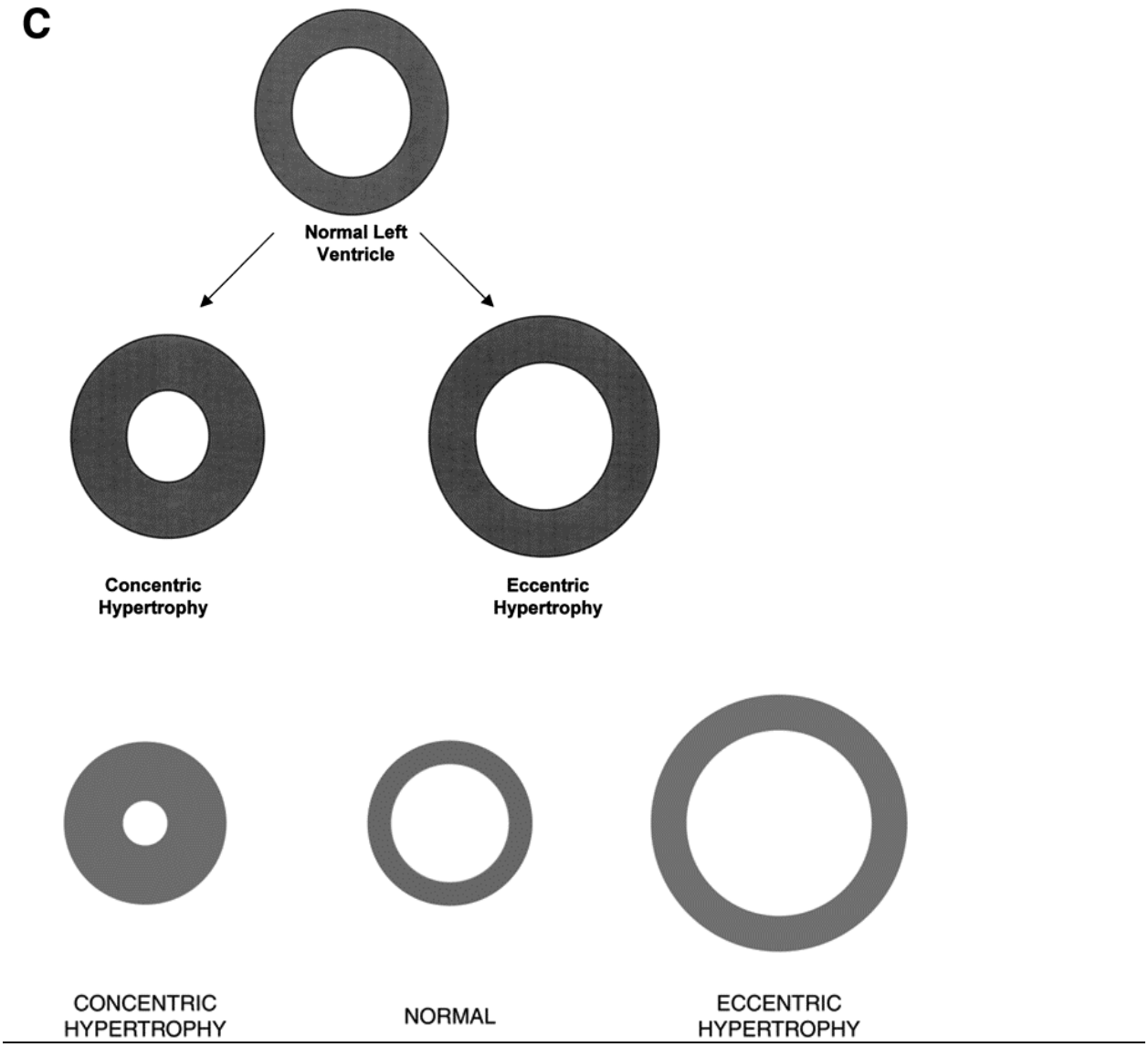


FIGURE 42

Concentric vs. eccentric hypertrophy.

Nadeau, J.H., et al., Pleiotropy, homeostasis, and functional networks based on assays of cardiovascular traits in genetically randomized populations. Genome Res, 2003. 13(9): p. 2082-91.

COMMENTARY

This is a simple figure illustrating physical deformation of a ventricle. The message is very simple – but is not optimally delivered due to the triangular layout of the ventricles. Can you tell whether the outer wall of normal and concentric ventricles are the same diameter? What about the inner wall of the normal and eccentric ventricles? By presenting all ventricles horizontally, and exaggerating the deformation, the characteristic difference between normal and the two abnormal conditions is made clear.

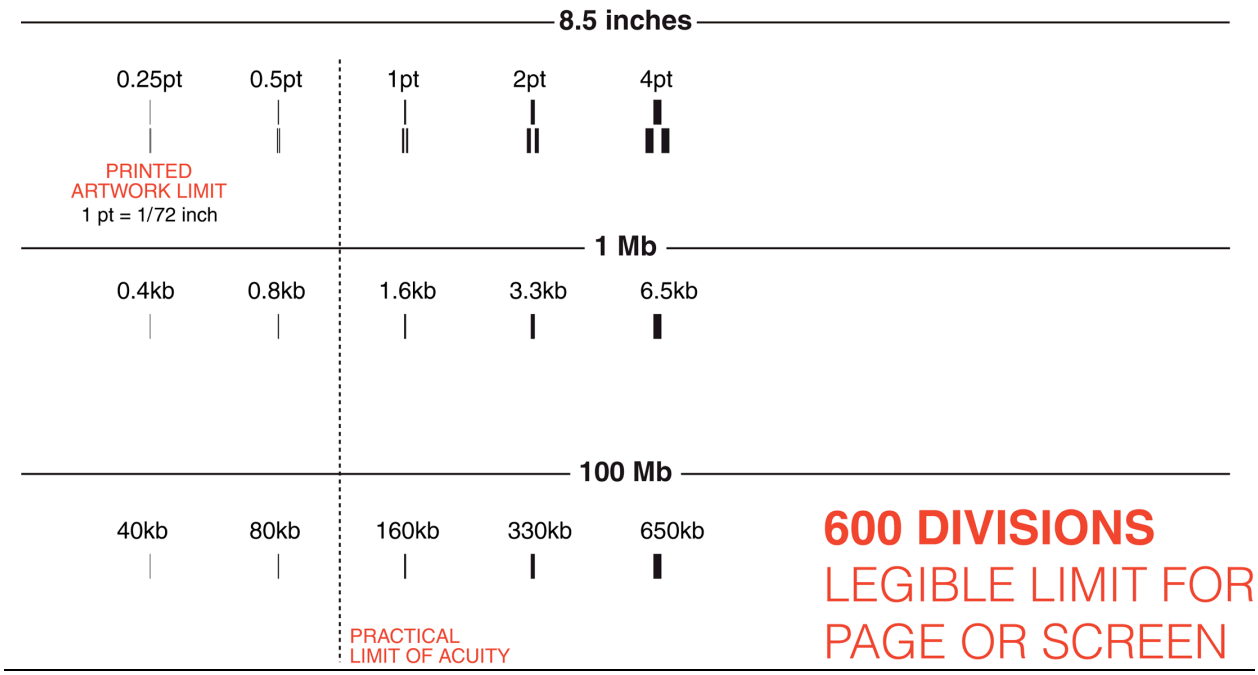


FIGURE 43

Distinguishing strokes narrower than 1pt (1/72th of an inch, 0.35mm) at an average reading distance is very difficult.

This limit of visual acuity places a more conservative limit on visualization than the resolution of the output device.

On a letter-size paper, a 1pt stroke corresponds to approximately 1.6kb if the page spans 1Mb and 160kb if the page spans 100Mb. The consequence of this is that if you are showing the entire human genome (with horizontal chromosomes stacked vertically), and need to accommodate 247Mb of chr1, the absolute minimum sample window for your figure should be about 400kb. For greater viewing comfort, you should multiple this by 4x, to yield a division of 1.2Mb.

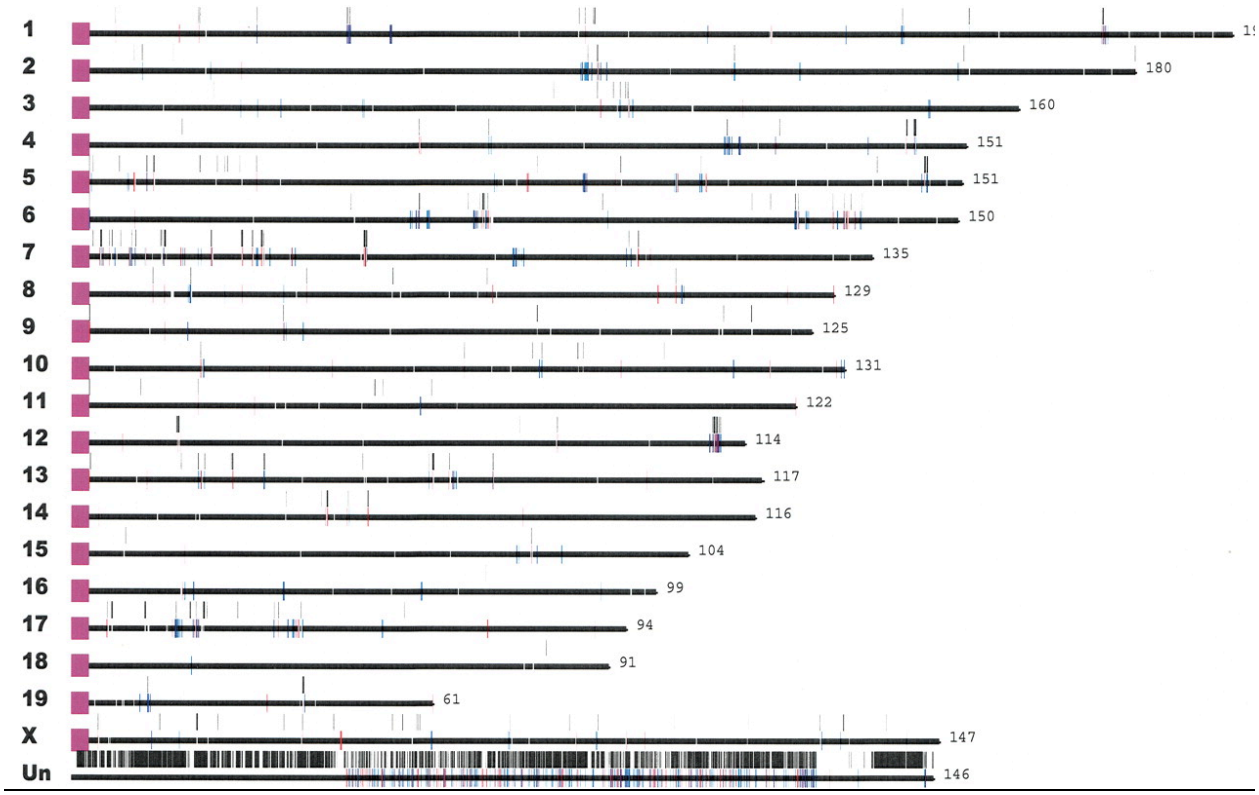


FIGURE 44

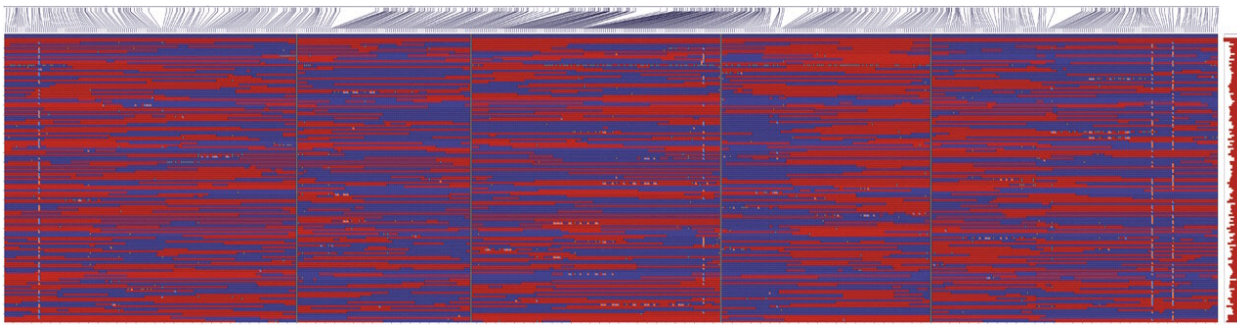
Mouse segmental duplications.

J. A. Bailey, D. M. Church, M. Ventura, M. Rocchi, E. E. Eichler, Genome Res 14, 789 (May, 2004).

COMMENTARY

Displaying small events on a whole-genome scale and keeping data visible is difficult. This figure shows individual segmental duplications on the mouse genome, but the size of the event is smaller than the resolution limit of acuity.

If distribution of events on the physical scale is important, events should be binned (e.g. every 1Mb) to show density. Otherwise, other statistical parameters (e.g. size, inter-event distance) should be reported instead.



zoom of top-right corner:

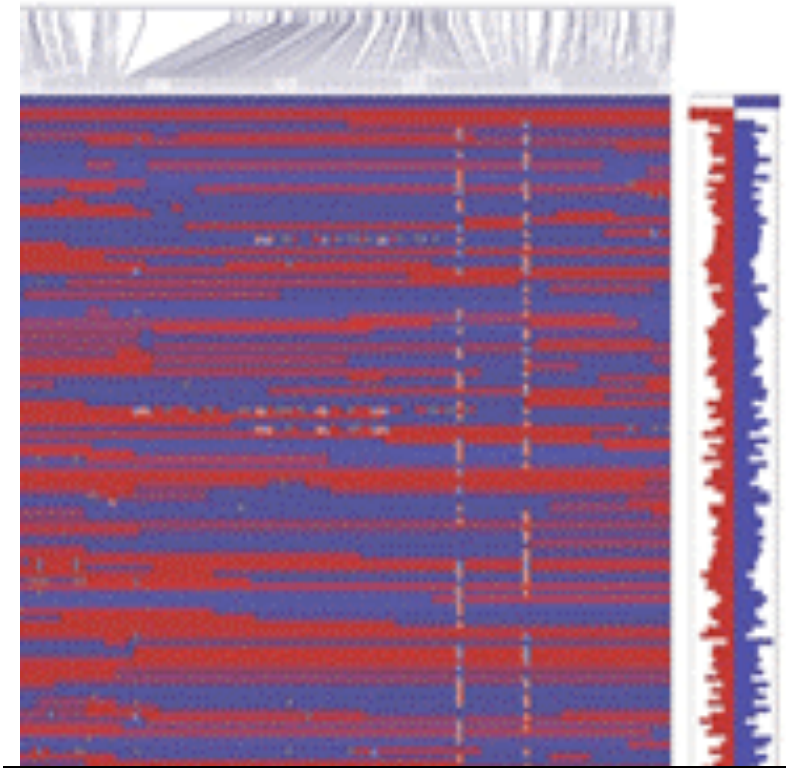


FIGURE 45

Haplotypes of 148 RILs plus parental genotypes.

West, M.A., et al., High-density haplotyping with microarray-based expression and single feature polymorphism markers in Arabidopsis. Genome Res, 2006. 16(6): p. 787-95.

COMMENTARY

The information in this figure is so dense, it appears random. Even when reproduced at very high resolution, the amount of information is overwhelming and the image is too large for practical navigation.

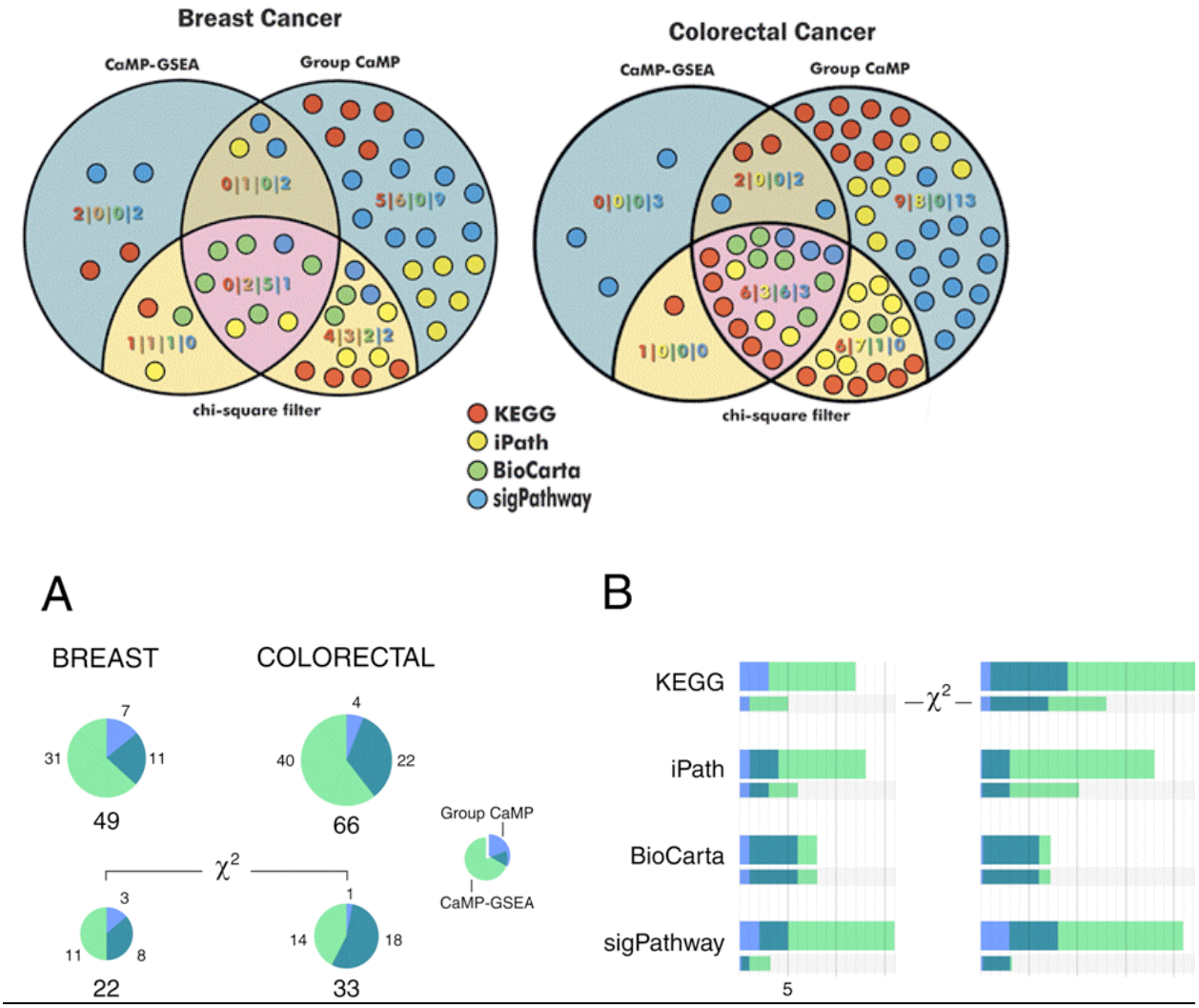


FIGURE 46

Comparison of mutation enrichment in cellular pathways using complementary statistical approaches.

Lin, J., et al., A multidimensional analysis of genes mutated in breast and colorectal cancers. Genome Res, 2007. 17(9): p. 1304-18.

COMMENTARY

This figure is too ambitious. It attempts to show data categorized by three independent variables: pathway, algorithm and presence of filter. Although the Venn approach may initially seem like a good choice (data points group into overlapping sets), the result is too complex. The figure also lacks consistency: note the orange label to indicate count of yellow circles (left) and the reuse of both blue and yellow to encode independent characteristics (yellow iPath, yellow filter Venn circles) (blue sigPathway, blue algorithm Venn circles). The redesign offers an entry point into the complex data set through panel A, setting up the color coding (note how the legend is used to communicate that the Venn segments overlap).

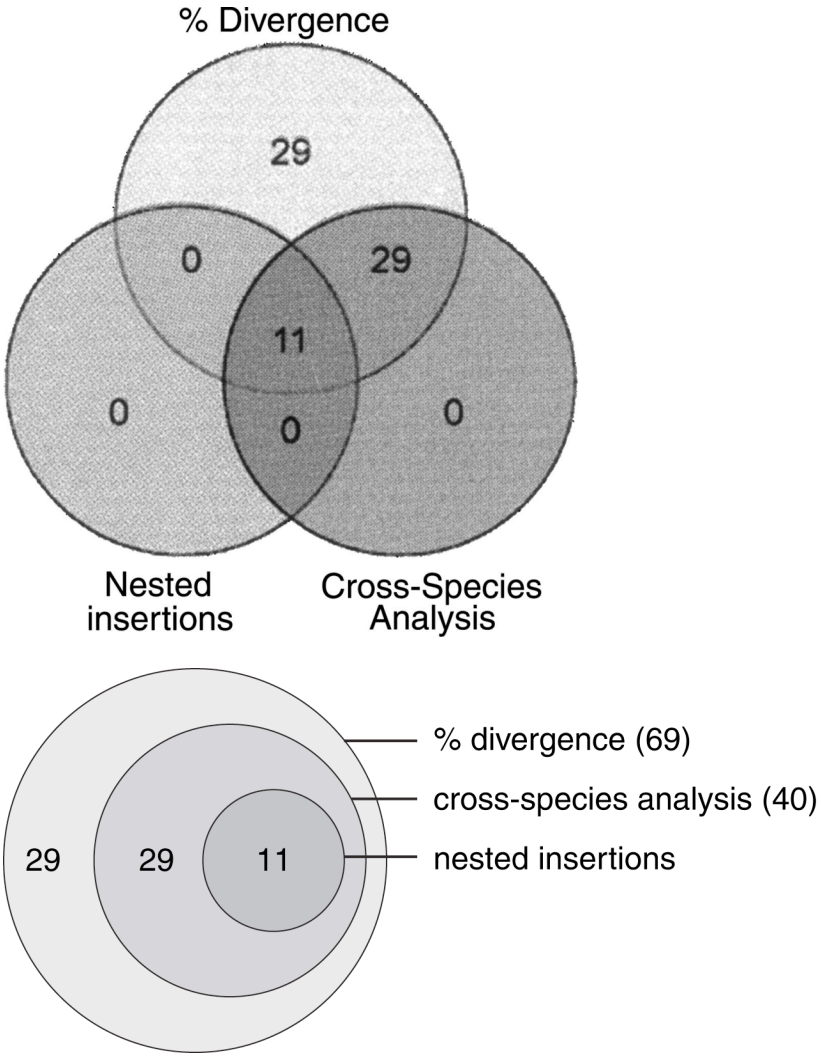


FIGURE 47

Comparison of three independent methods for dating DNA transposons.

Pace, J.K., 2nd and C. Feschotte, The evolutionary history of human DNA transposons: evidence for intense activity in the primate lineage. Genome Res, 2007. 17(4): p. 422-32.

COMMENTARY

Whereas a Venn diagram was inapropriate in Figure 46 because the data set was too complex, it is inappropriate here because the data is too simple.

How long did it take you to determine that the Venn diagram actually represented proper subsets? The redesigned version shows this immediately.

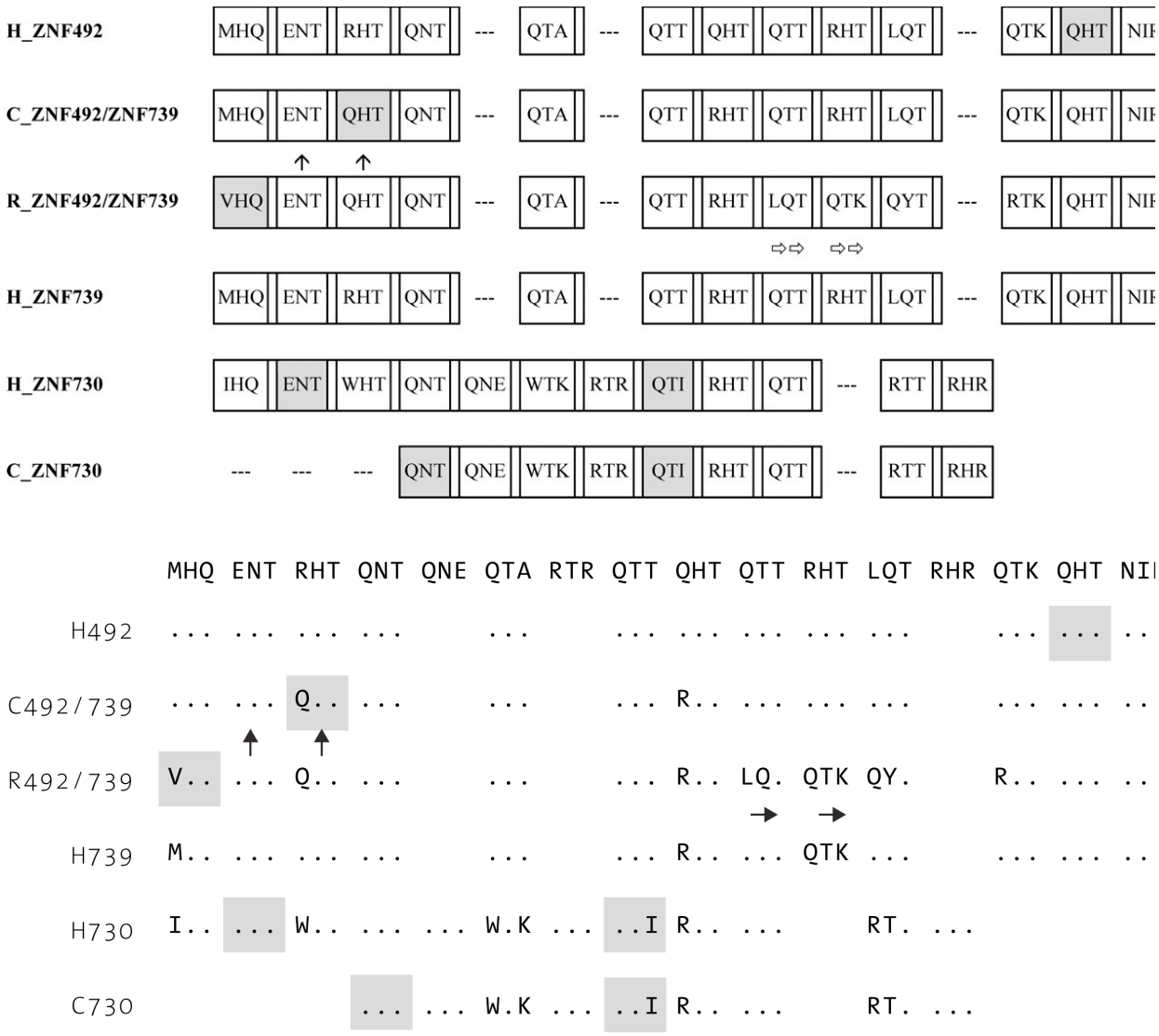


FIGURE 48

Zinc finger exon analysis for ZNF493 and ZNF738, two divergent genes from the ZNF431 clade.

Hamilton, A.T., et al., Evolutionary expansion and divergence in the ZNF91 subfamily of primate-specific zinc finger genes. Genome Res, 2006. 16(5): p. 584-94.

COMMENTARY

This visualization suffers from an excess of ink. Too much of the same information is repeated, making it difficult to quickly identify differences.

By placing the consensus sequence on the top row, and using "." to indicate an unchanged amino acid, it is immediately clear which proteins are different, and where.

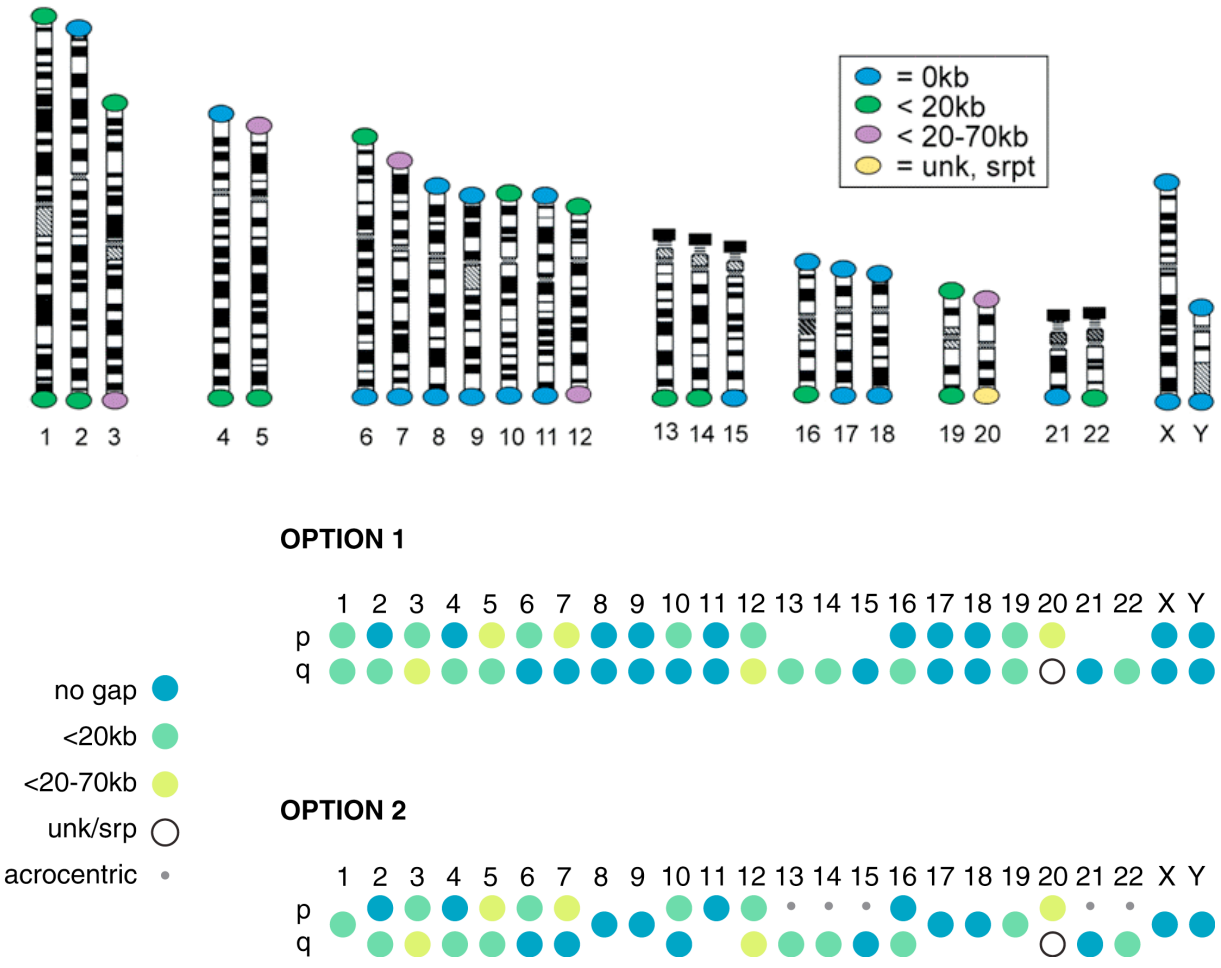


FIGURE 49

Telomere sequence gaps.

Riethman, H., et al., Mapping and initial analysis of human subtelomeric sequence assemblies. Genome Res, 2004. 14(1): p. 18-28.

COMMENTARY

Like the figure above, the majority of this figure is dedicated to content that is not relevant to the information being communicated. The ideogram structure of the chromosomes is not important in this context, which presents categorization of telomeres.

The redesign uses a minimalist approach. Notice that in Option 1, acrocentric chromosomes have no entries for p telomeres. It is clear that these telomeres cannot be characterized, because there is an independent category for “unknown”, which is used for 20q. To quickly identify which chromosomes have the same telomere gaps, this case is encoded by a single glyph in Option 2. Note that in this case a separate symbol is used for acrocentric chromosomes to help distinguish chromosome entries that use one glyph.

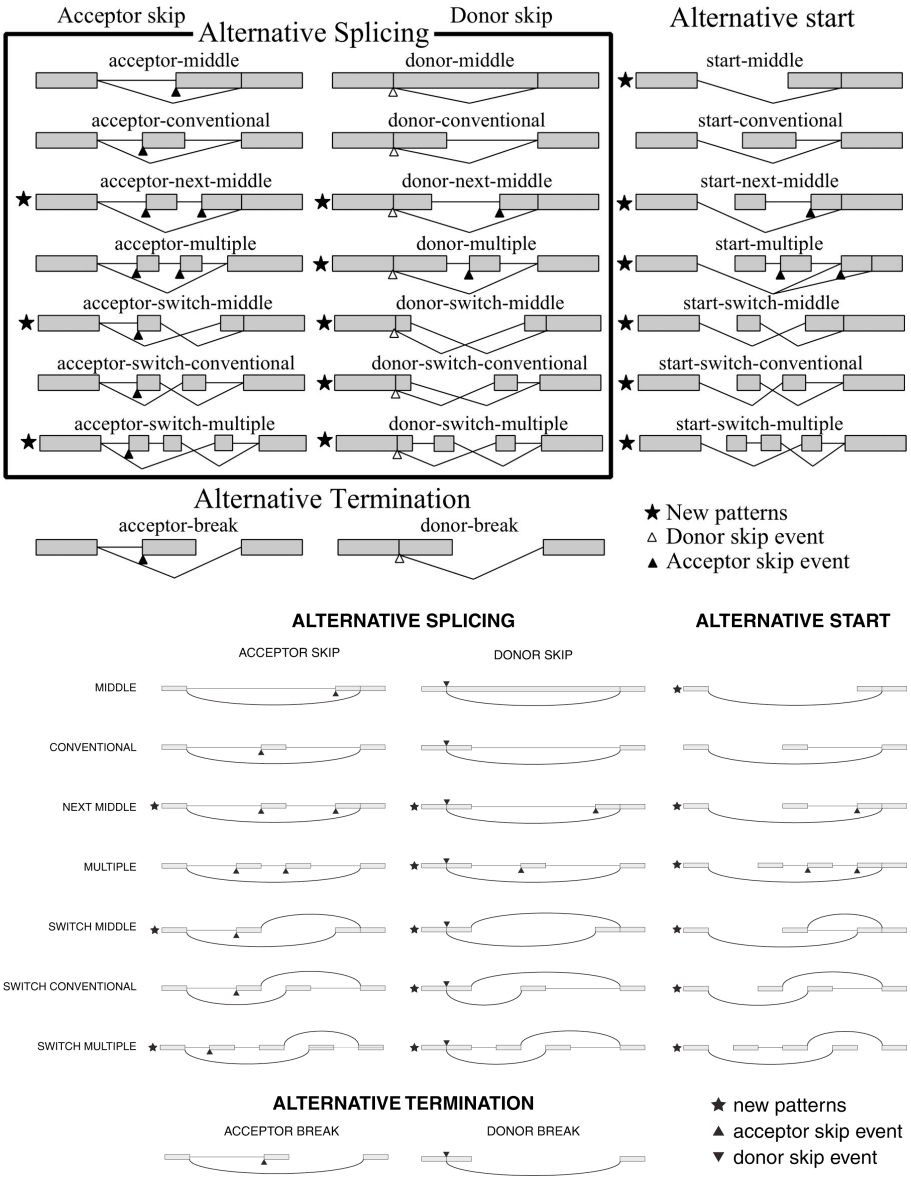


FIGURE 50

Combinatorial classification of ATS units.

Sharov, A.A., D.B. Dudekula, and M.S. Ko, Genome-wide assembly and analysis of alternative transcripts in mouse. Genome Res, 2005. 15(5): p. 748-54.

COMMENTARY

Splicing diagrams are notorious for cart junk – decorative and unnecessary elements. This figure focuses too much on the exons making it impossible to identify any pattern in the splicings. Moreover, significant portion of text in labels is duplicated, making it hard to quickly identify categories. The redesign uses curves for splicing and subtle rectangles for exons, which helps to distinguish these two features. Each exon is made to be the same length and exon boundaries are vertically aligned to maintain a grid structure. The number of labels is reduced by decomposing them into independent categories.

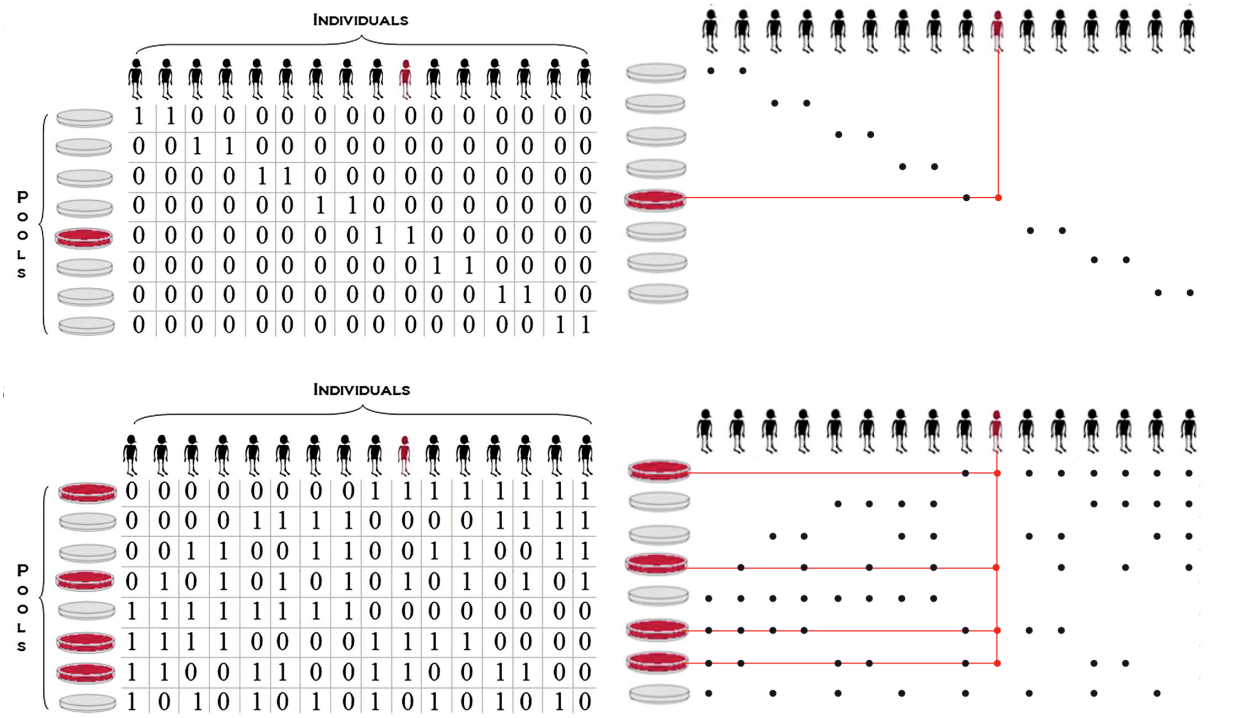


FIGURE 51

Resequencing with naïve and log pool designs.

Prabhu, S. and I. Pe'er, Overlapping pools for high-throughput targeted resequencing. Genome Res, 2009. 19(7): p. 1254-61.

COMMENTARY

By explicitly encoding the absence of a sample in a pool with 0, the figure obscures the pattern of 1s, which is the important element. Typically, when encoding binary data it is not necessary to have a symbol for both 0 and 1 states. A blank for 0 is sufficient. Similarly, missing data (there are many reasons why a data point might be missing) should be encoded minimally. Do not use any ink (or as little as possible) to encode missing data.

The redesign makes the relationship between individuals and pools explicit. The red horizontal and vertical guides draw attention to the combination of pools containing the selected individual's sample. Notice that the labels "individuals" and "pools" are not necessary, since these are graphically encoded in the images in the table headers.

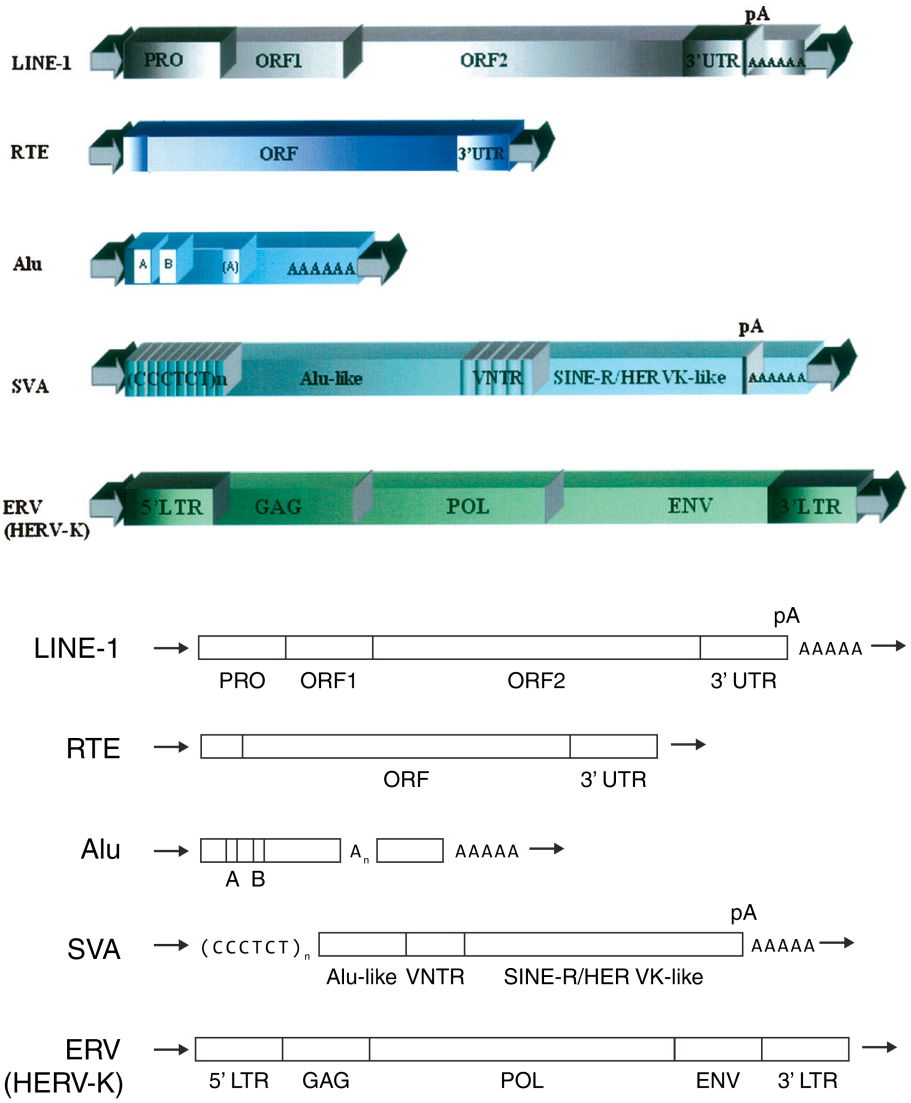


FIGURE 52

Schematic representation of the genome organization of mammalian retroelements.

Belancio, V.P., D.J. Hedges, and P. Deininger, Mammalian non-LTR retrotransposons: for better or worse, in sickness and in health. Genome Res, 2008. 18(3): p. 343-58.

COMMENTARY

Ornamentation detracts from clear communication. It is hard to see through the 3D boxes and identify any patterns. Notice that sequence and encoded in the same way as a region category – $(CCCTCT)_n$ and Alu-like are given the same symbol, which confounds the visual grammar.

The redesign distinguishes sequence content from categories. One can argue that the arrows at the start and end of each retroelement are not important, since they appear for each.

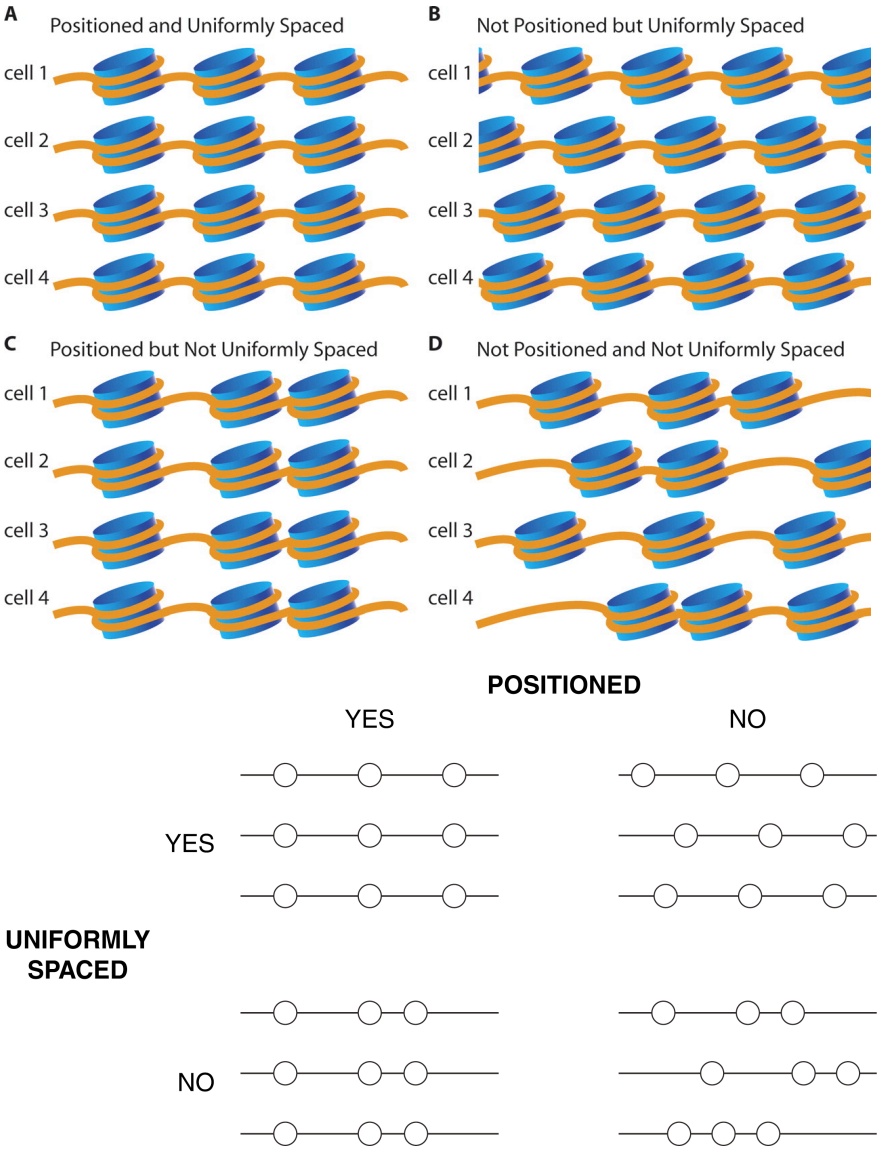


FIGURE 53

Possible patterns of nucleosome positioning.

Valouev, A., et al., A high-resolution, nucleosome position map of C. elegans reveals a lack of universal sequence-dictated positioning. Genome Res, 2008. 18(7): p. 1051-63.

COMMENTARY

Similar to the splicing and retroelement figures above, this figure focuses on an unimportant part of the data. The authors intend to communicate the difference in spacing between nucleosomes, but hide this information by overpowering the image with a visually strong representation for the nucleosome. Is it important to depict the nucleosomes realistically? By simplifying the nucleosomes down to a circle, the redesigned figure clearly demonstrates the spacing. By presenting the figure in the form of a table, with the *positioned* and *uniform* categories as headers, the two-parameter classification is instantly recognized.

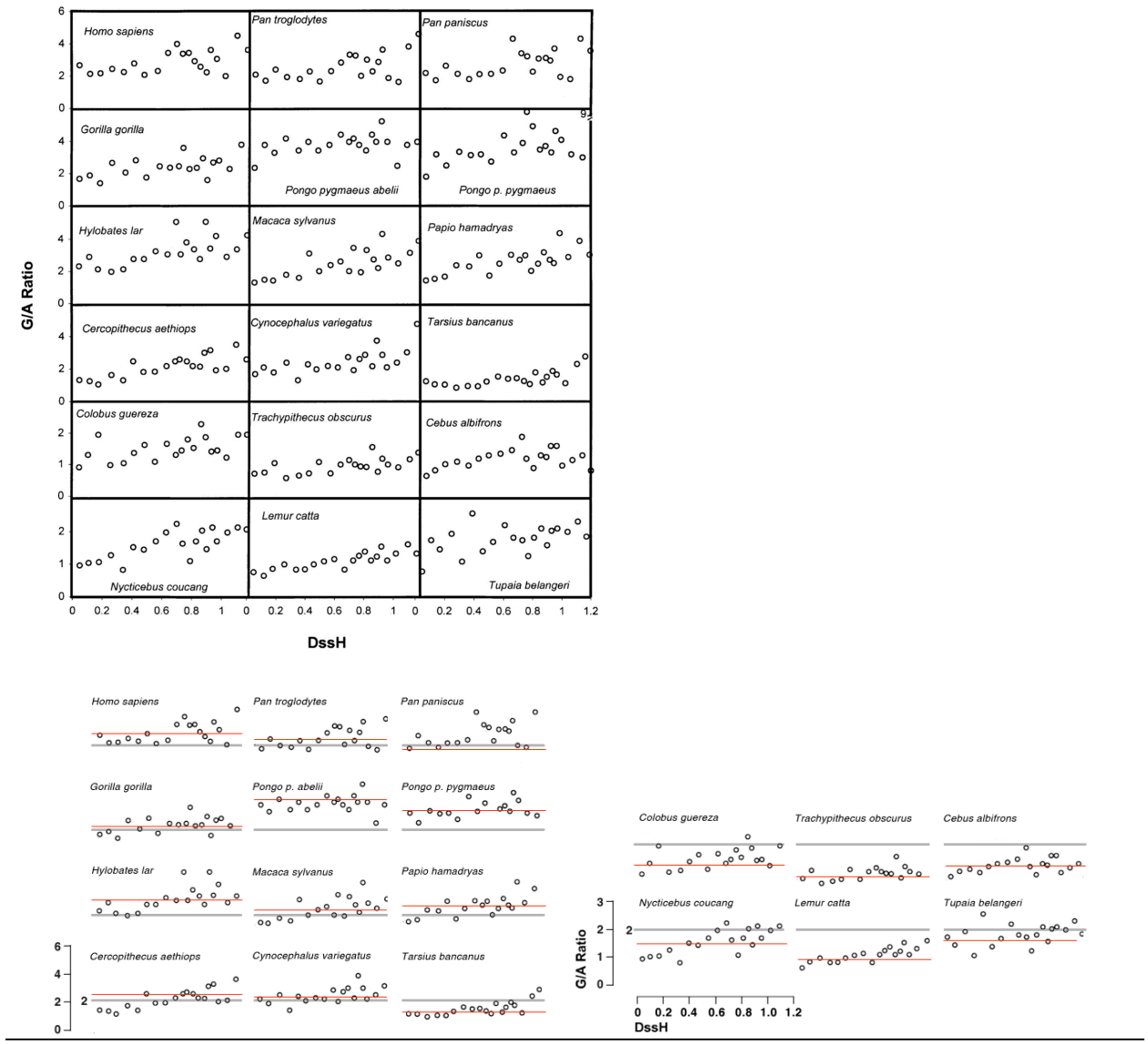


FIGURE 54

G/A ratios for complete primate mitochondrial genomes and two near outgroups.

Raina, S.Z., et al., Evolution of base-substitution gradients in primate mitochondrial genomes. Genome Res, 2005. 15(5): p. 665-73.

COMMENTARY

When a panel of plots its shown, it is not usually necessary to repeat the axes. Even if the scales are different (you should have a good reason to *not* keep the axis ranges fixed), the dominance of the axes and borders on the page can outweigh the data themselves. In the redesign a single reference (at G/A=2) is chosen for each data set, and a horizontal average is identified with a thin red line, making comparison easy. Species labels are placed on a grid – this alignment limits eye travel. Ideally, the order of data panels should correspond to a meaningful property of the data set (e.g. average, variation, etc), so that each species can be quickly compared.

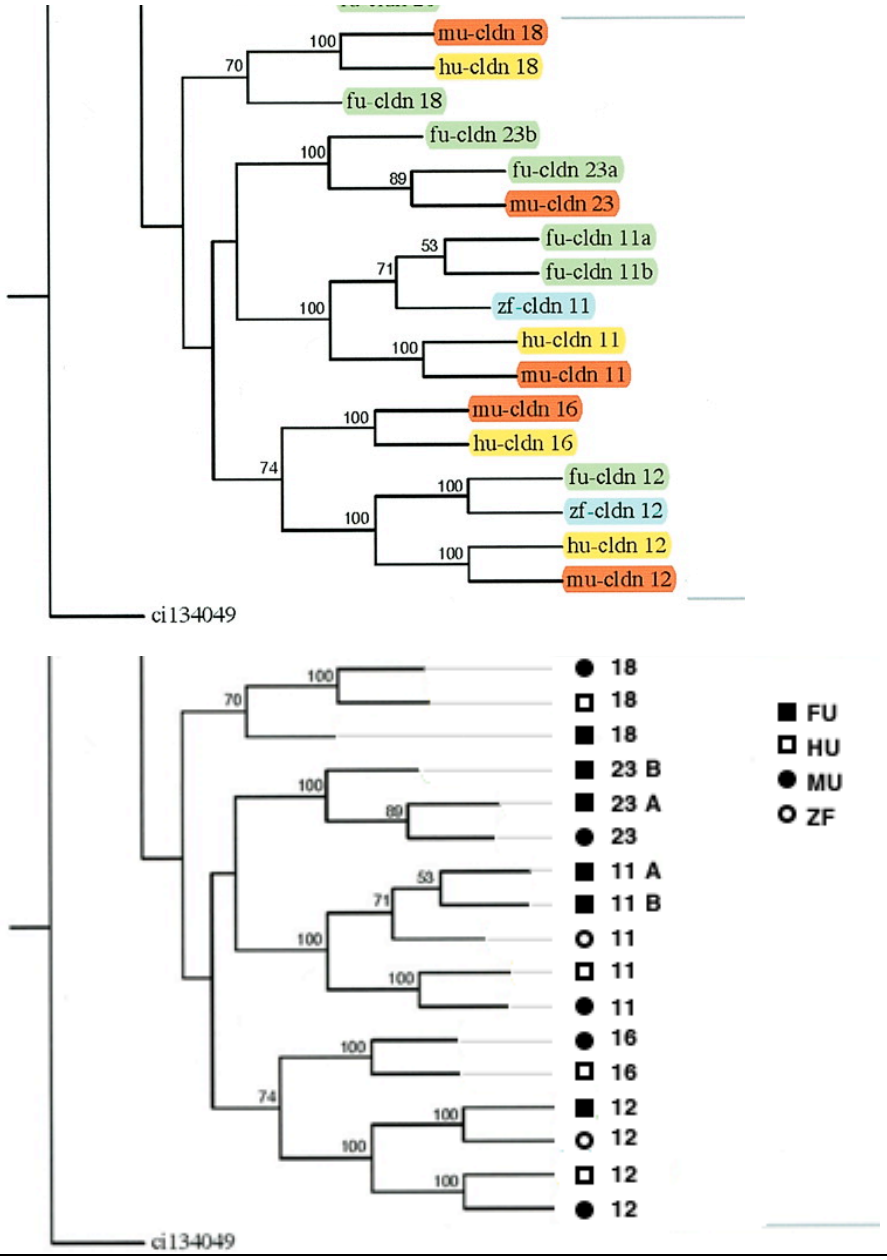


FIGURE 55

Consensus phylogenetic tree of Claudin proteins.

Y. H. Loh, A. Christoffels, S. Brenner, W. Hunziker, B. Venkatesh, Genome Res 14, 1248 (Jul, 2004).

COMMENTARY

The labels for each node contain redundant information. Since each label contains "-cldn", this content can be removed. It is much easier for the reader to see the same information once, rather than repeated, since it requires a complete examination of the entire figure to satisfy oneself that the information is indeed the same.

By aligning the labels in the redesign it is easier to scan down and quickly compare both categories and numbers. Presence of suffixes is immediately obvious (23B vs 23).

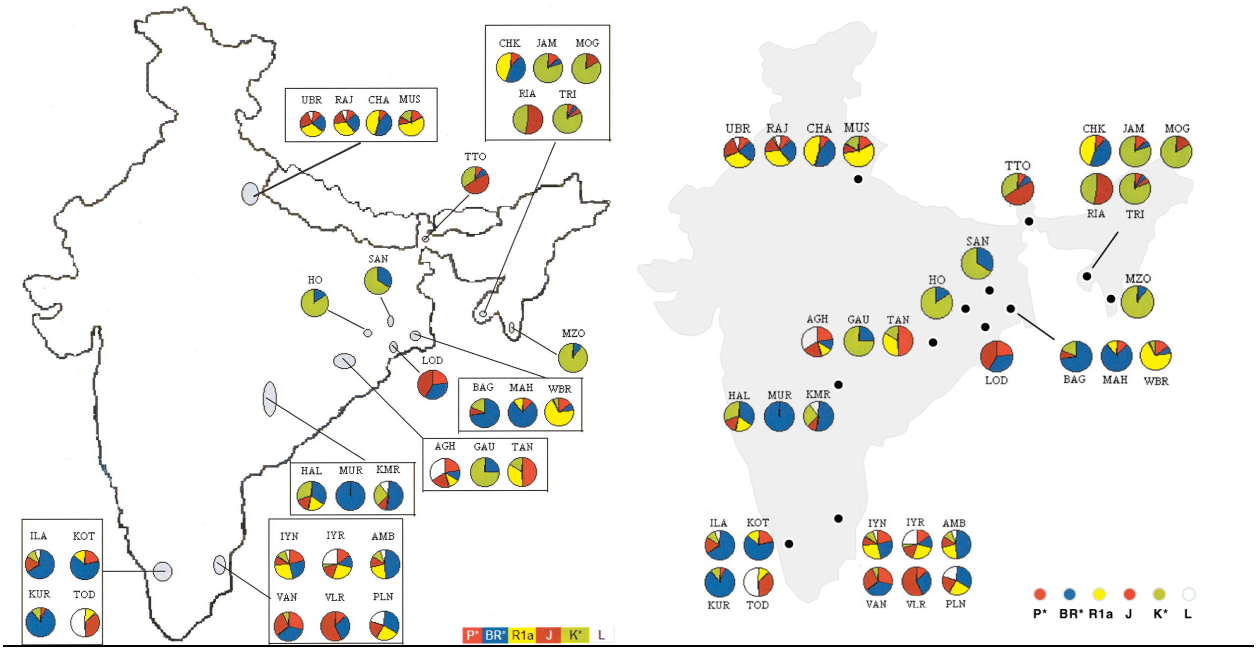


FIGURE 56

Frequencies (%) of Y-chromosomal haplogroups among ethnic populations.

Basu, A., et al., Ethnic India: a genomic view, with special reference to peopling and structure. Genome Res, 2003. 13(10): p. 2277-90.

COMMENTARY

This figure effectively focuses on the data, but contains extraneous elements (e.g. shapes of regions pointed to by the pie charts are all different – this is not important) and lacks visual organization.

The redesign uses a grid structure on which the pie charts are placed, which immediately helps to organize the relationship between regions and pie charts. In fact, only two link lines are needed to relate distantly placed pie charts to their regions. Note that both link lines are oriented at 45 degrees to the horizontal. Keeping the same length and angle (or family of angles) for link lines makes the final figure tidy.

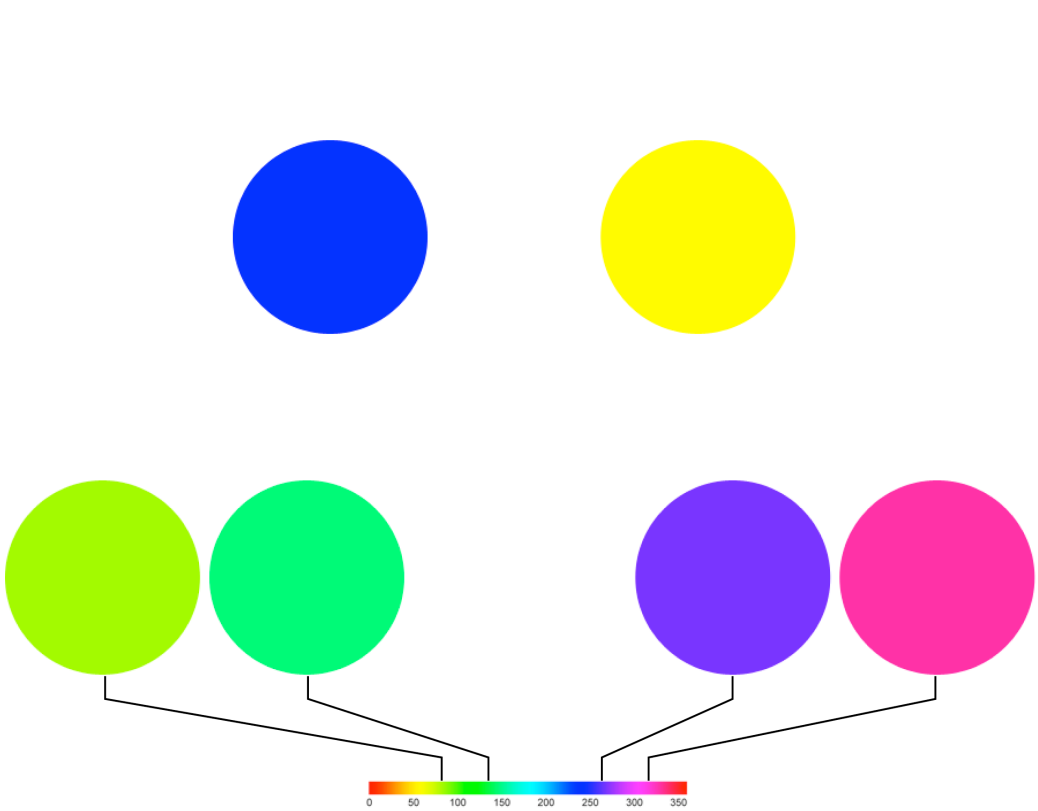


FIGURE 57

The perception and characterization of color can be discrepant in color spaces that are not *perceptually uniform*.

Both the blue and yellow color in (A) have the same saturation and value in HSV space (hue, saturation, value), only varying by hue. Yet, the yellow is perceived to be significantly brighter. This difference in perception is not reflected in the relative position of the colors in HSV space.

The two pairs of colors in (B) are both composed of colors that vary by 60° (1/6 of the color wheel), yet the two green in the left pair appear to be much more similar than the colors in the right pair. The perceived difference in the two colors is not proportional to their distance in HSV.

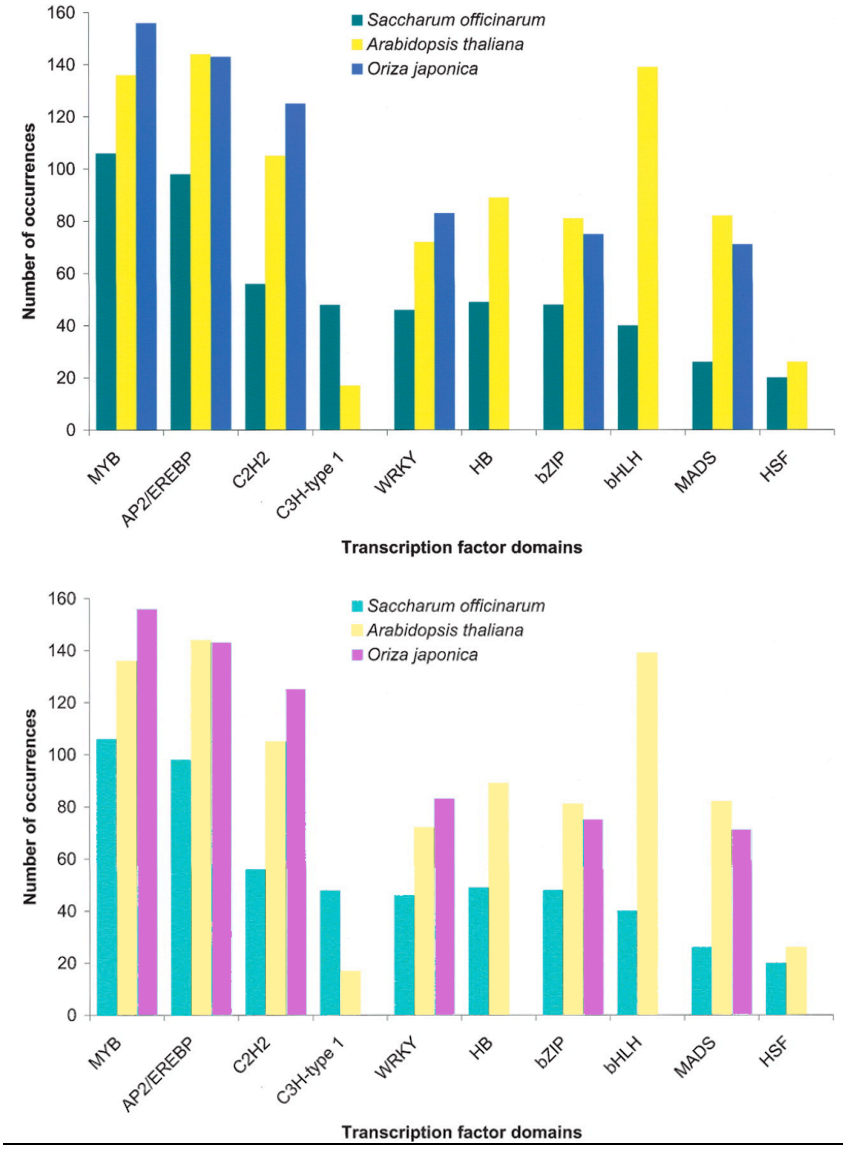


FIGURE 58

The 10 most common transcription factor Pfam domains in SAS proteins.

Vettore, A.L., et al., Analysis and functional annotation of an expressed sequence tag collection for tropical crop sugarcane. Genome Res, 2003. 13(12): p. 2725-35.

COMMENTARY

Perceptual characteristics of color should be taken into account when selecting a color palette. The yellow in this figure captures attention, drawing it away from the other two bin categories. This is due to the fact that pure yellow is perceived as very bright in comparison to other pure colors, such red, which appears much darker.

In the redesign the color scheme has been altered to normalize the luminance (perceived brightness) of the colors. The yellow no longer competes with the other colors.

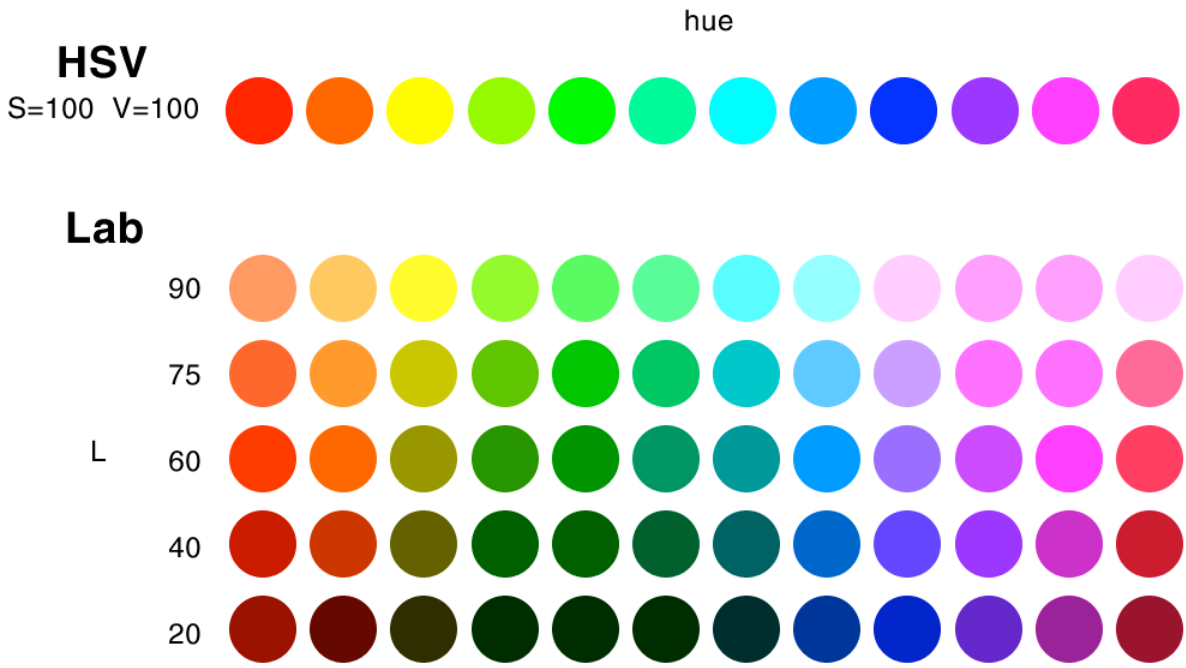


FIGURE 59

By using a color space that takes perception into account (e.g. LAB or the more intuitive LCH space which is a perceptually uniform HSV equivalent), the difference in colors can be limited to hue only.

Bottom rows in the figure represent colors that are normalized to have the same luminance. Note how each color row appears significantly more perceptually uniform than the unnormalized counterpart.

FIGURE 60

Conventional human chromosome color assignment used by UCSC Genome Browser. These colors are defined in `color.conf` as `chr1`, `chr2`, `chr3`, etc.

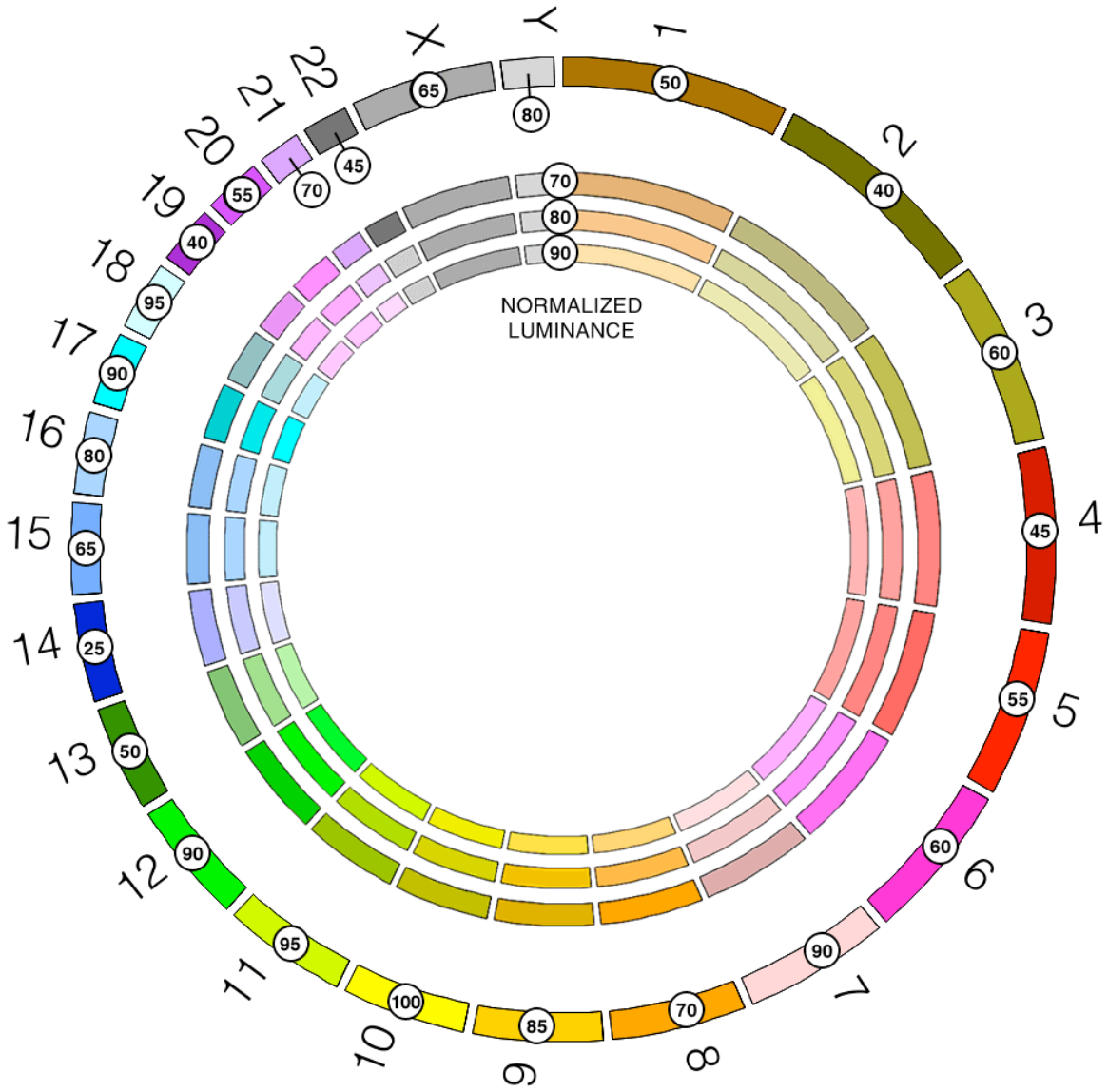


FIGURE 61

The result of applying luminance normalization (Figure 59) to the conventional human color assignment (Figure 60). The conventional palette has colors that range in luminosity from 23 (chr14) to 98 (chr10). Sets of normalized equivalents are shown for luminances 70, 80 and 90. The result is a more harmonious scheme, with colors remaining distinguishable.

The luminosities in this figure are shown rounded off to the nearest 5.

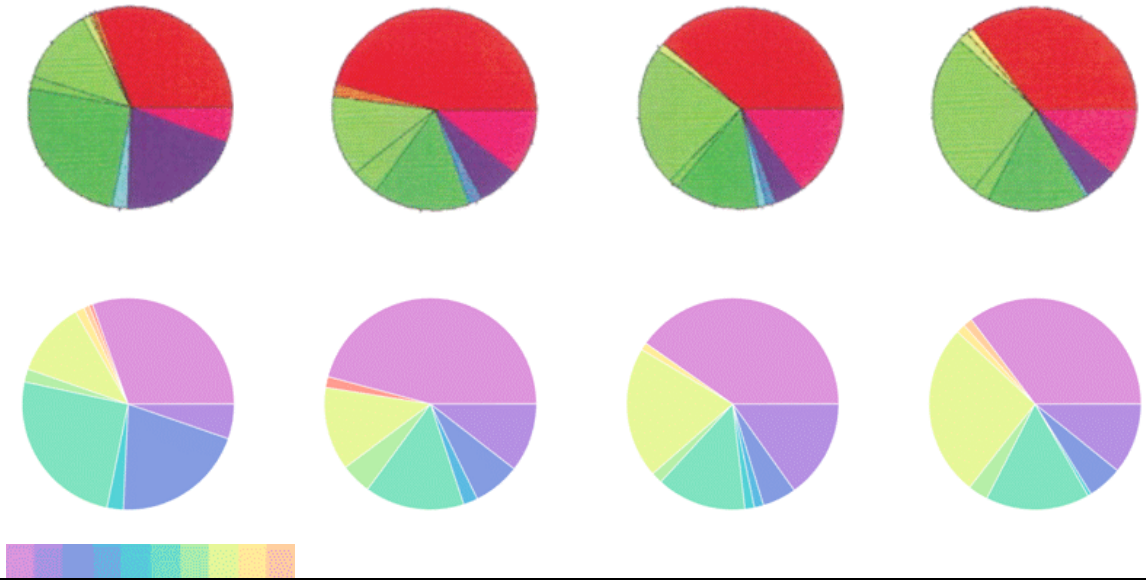


FIGURE 62

Pie charts for tissue profiling by Gene Ontology.

Bono, H., et al., Systematic expression profiling of the mouse transcriptome using RIKEN cDNA microarrays. Genome Res, 2003. 13(6B): p. 1318-23.

COMMENTARY

Multi-color palettes are especially difficult to design because balancing desired perceptual properties of multiple colors (equal perceived distance, equal perceived importance, and, if required, naturally perceived order) is hard. Luckily a large number of palettes have already been created for this purpose. These are the Brewer palettes (*www.colorbrewer.org*), which address the problem of encoding qualitative, diverging and sequential data categories.

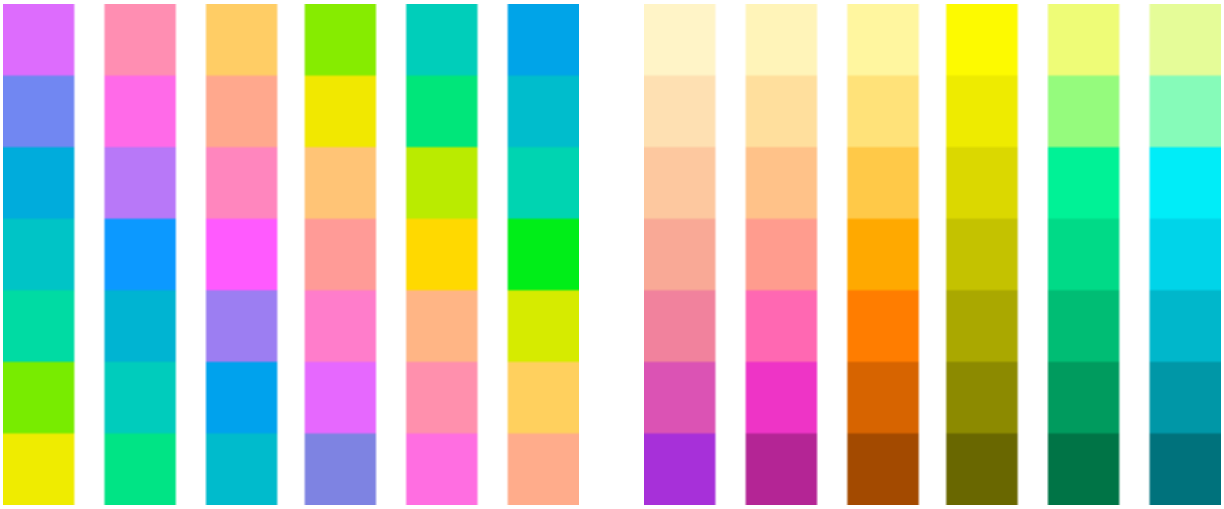


FIGURE 63

Examples of qualitative (left) and sequential (right) 7-color Brewer palettes.

Qualitative palettes have an equal perceived importance and distance between colors. Sequential palettes add a natural order to the colors.

www.colorbrewer.org

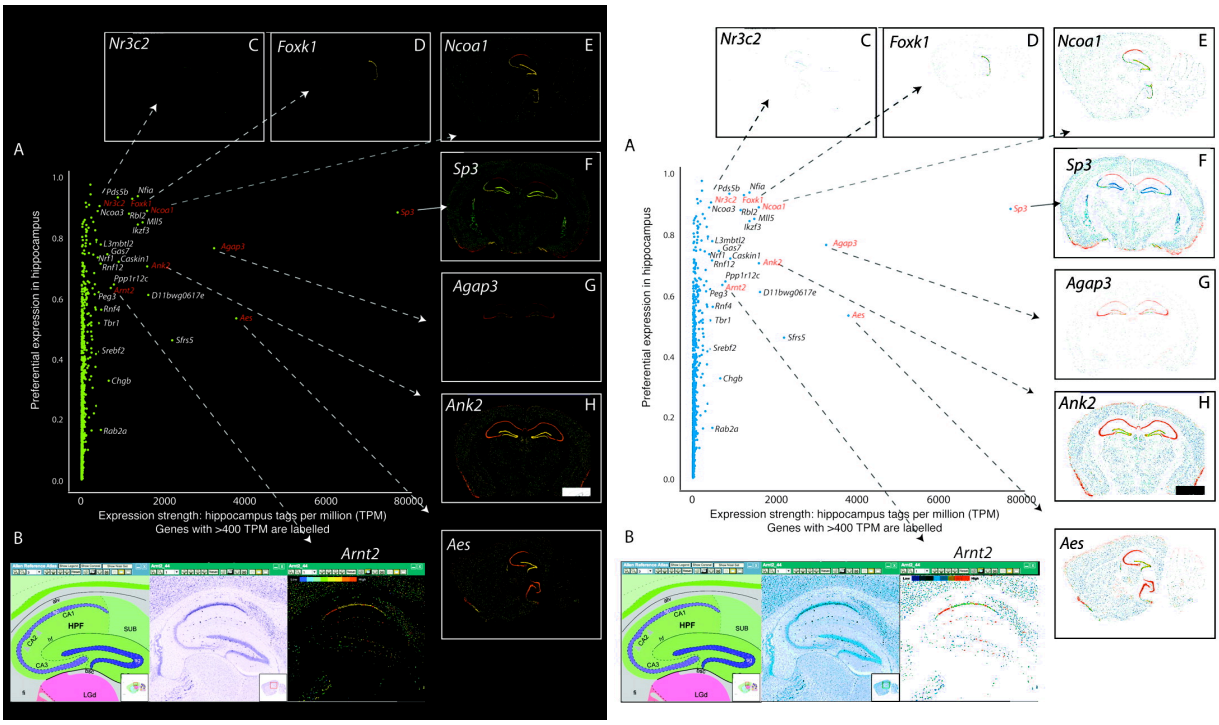


FIGURE 64

Transcription factor genes with preferential expression in hippocampus.

Valen, E., et al., Genome-wide detection and analysis of hippocampus core promoters using DeepCAGE. Genome Res, 2009. 19(2): p. 255-65.

COMMENTARY

Just as a choice of color palette is crucial to communicating texture in the data, the choice of background color is equally, if not more, important.

A dark background can entirely hide data, such as in this figure. By simply inverting the image and adjusting contrast, brain cross-sections in individual panels are visible.