Circos

round is good

Martin Krzywinski

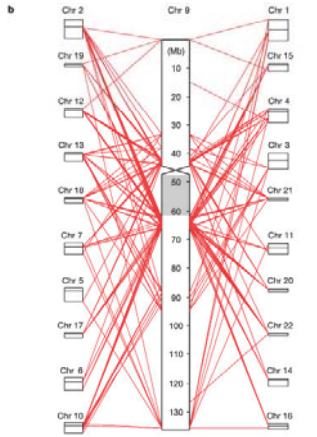
martin@bcgsc.ca

http://mkweb.bcgsc.ca/circos

What is Circos?

circos

- Circos makes drawing certain kinds of data easier and produces meaningful images that make data interpretation easy
- Circos is ideally suited for imaging relationship between positional data
  - a relationship between two locations on an integer line (e.g. a chromosome)
  - a relationship between two objects in a set
- by compositing the axes circularly, instead of along straight lines, relationship views become less cluttered



Humphray, S. J., K. Oliver, et al. (2004). "DNA sequence and analysis of human chromosome 9." Nature 429(6990): 369-74.

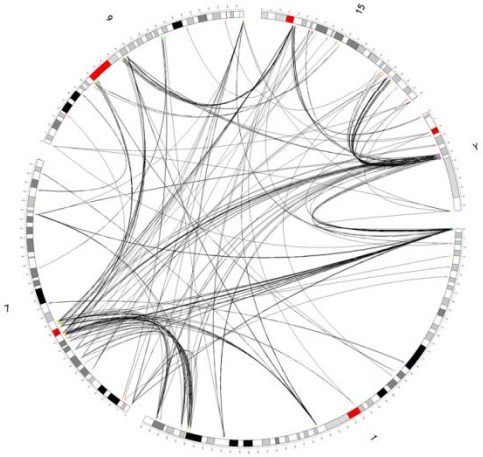


image
by Circos

instead of this → how about this?

Focus on Genomic Data

circos

- since I work in genomics, I have spent most of my time applying Circos to data in this field, but circular axis layout can be applied to visualizing other data (e.g. database table relationships)
- this talk will focus on genomics, though

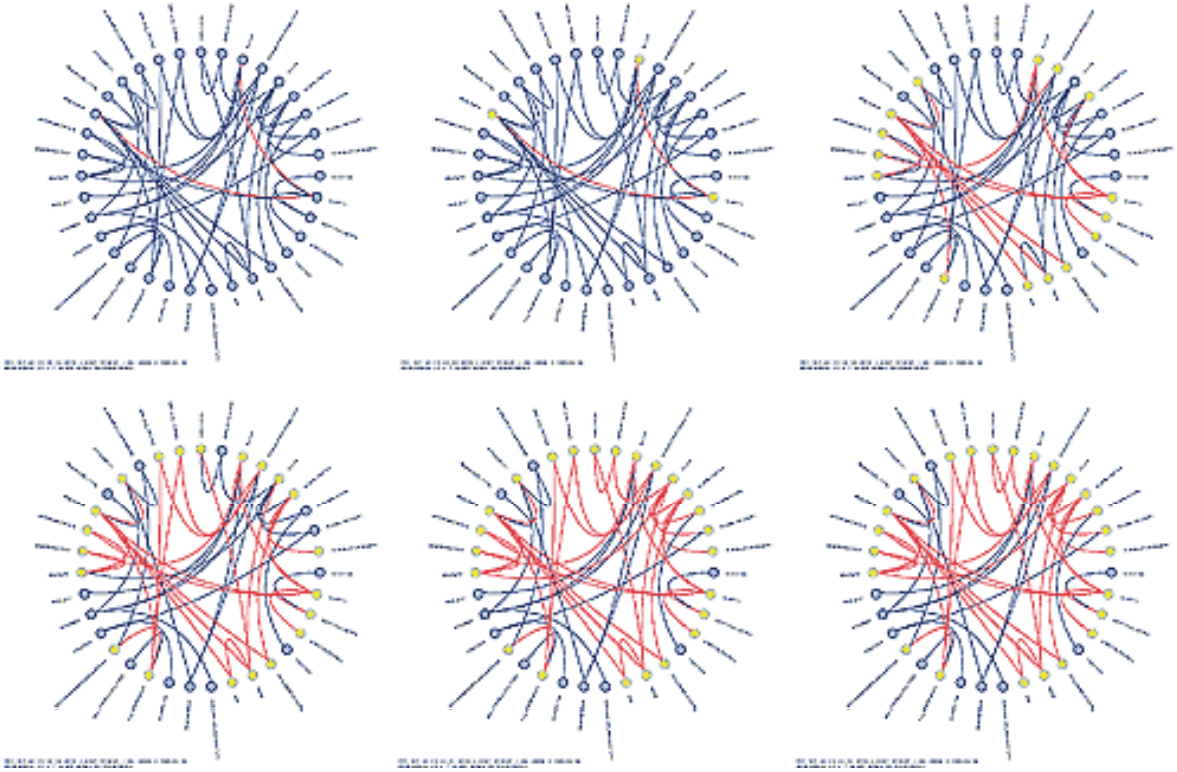


image by Schemaball shows foreign key relationships between tables in a database

here, each glyph along the circle represents a table, and joining lines represent foreign keys

mkweb.bcgsc.ca/schemaball

Why Reinvent the Wheel – Another Browser?

circos

- there are many genome browsers already available – do we really need another? U
  - UCSC genome browser (genome.ucsc.edu)
  - Ensembl (ensembl.org)
  - Vista (pipeline.lbl.gov/cgi-bin/gateway2)
  - VEGA (vega.sanger.ac.uk)
  - ARGO (www.broad.mit.edu/annotation/argo)

- I think we do, to draw data structures that obfuscate common diagram formats
  - standard 2D plots (2 perpendicular axes) are inadequate for data that relate two genomic positions (e.g. alignments, conservation)
  - a custom axis layout (e.g. circular, like in Circos) can help

- communicating data visually is critical for large data sets
  - very applicable to genomics, where positional features (e.g. genes) are much smaller than the data domain (e.g. chromosome)
  - particularly important when data sets are complex, with latent patterns

Types of Data Relationships

circos

- in a general sense, data is either scalar or vector, and mappings between data are either scalar, or vector valued
- the genome is a 1-dimensional data structure – a genomic position is thus a scalar

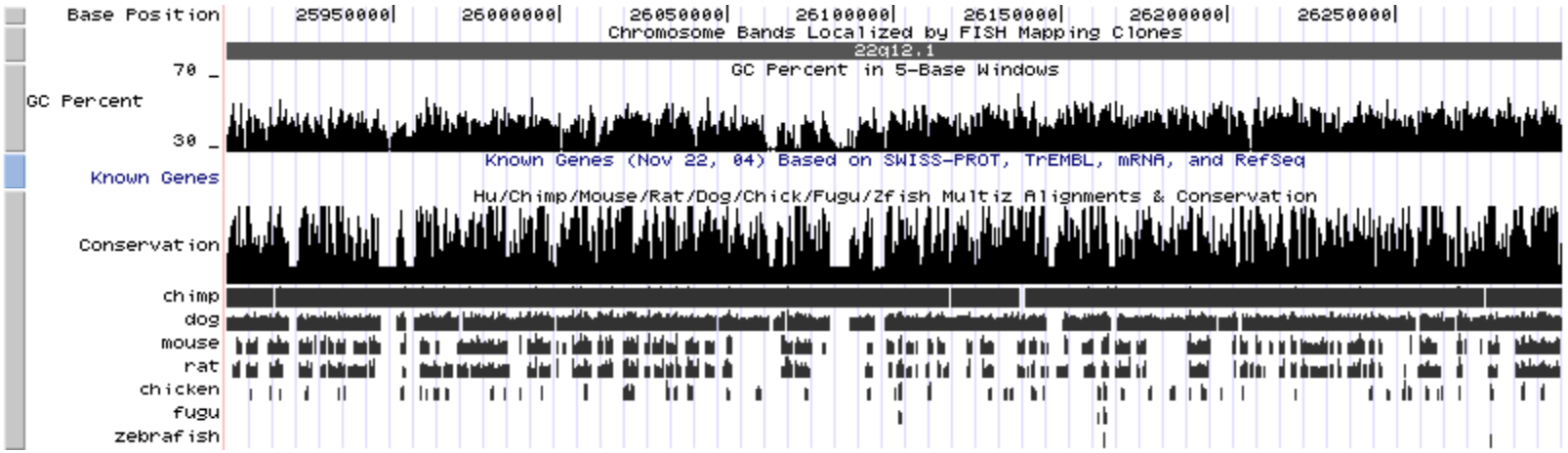
		output: scalar	output: vector
input	scalar	**GC content, coverage** scatter, line, histogram $f : g \rightarrow y$	**alignments (duplications, synteny)** **end sequence alignments, clone mappings** colour map, ideograms connected by lines, tilings $f : g \rightarrow [(g', v)_1, (g'', v)_2, \ldots]$
	vector	**alignment identity (duplications, synteny)** dot plot, colour map, surface/solid plot $f : [g, g', \ldots] \rightarrow y$	**generalized alignments** hard $f : [g, g', \ldots] \rightarrow [(g', v)_1, (g'', v)_2, \ldots]$

Scalar to Scalar Mappings

circos

- scalar valued mappings are very common and easily handled
  - input genomic position is a scalar input
  - when the output is real-valued (GC content, degree of conservation, etc) use a histogram, line plot, scatter plot
    - genome position on x-axis
    - function value on y-axis
  - this works very well when the dynamic range of the range is much smaller than the domain

$$f : g \rightarrow y$$

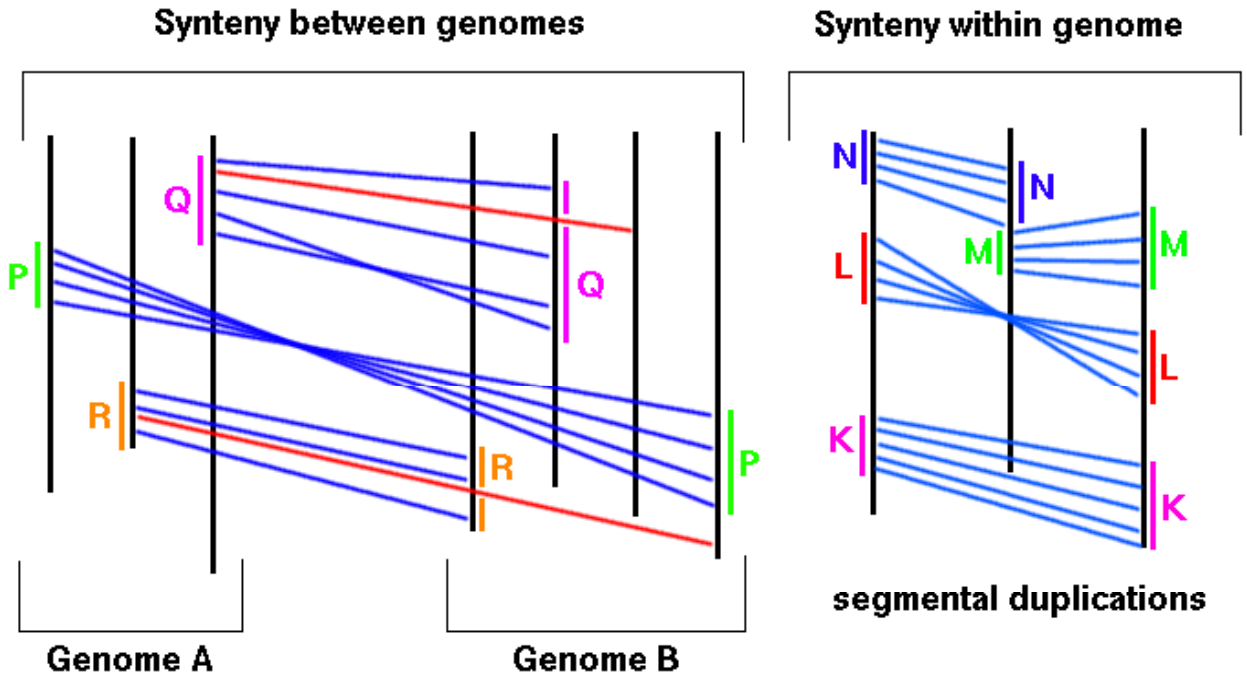


UCSC Genome Browser (hg17)

Scalar to Scalar Mappings

circos

- trouble arises when the output scalar is also a genome position
  - range may be the same genome, or a different genome
  - in this case, the dynamic range of the domain is comparable to the range (3Gb-to-3Gb)



$$f : g \rightarrow g'$$

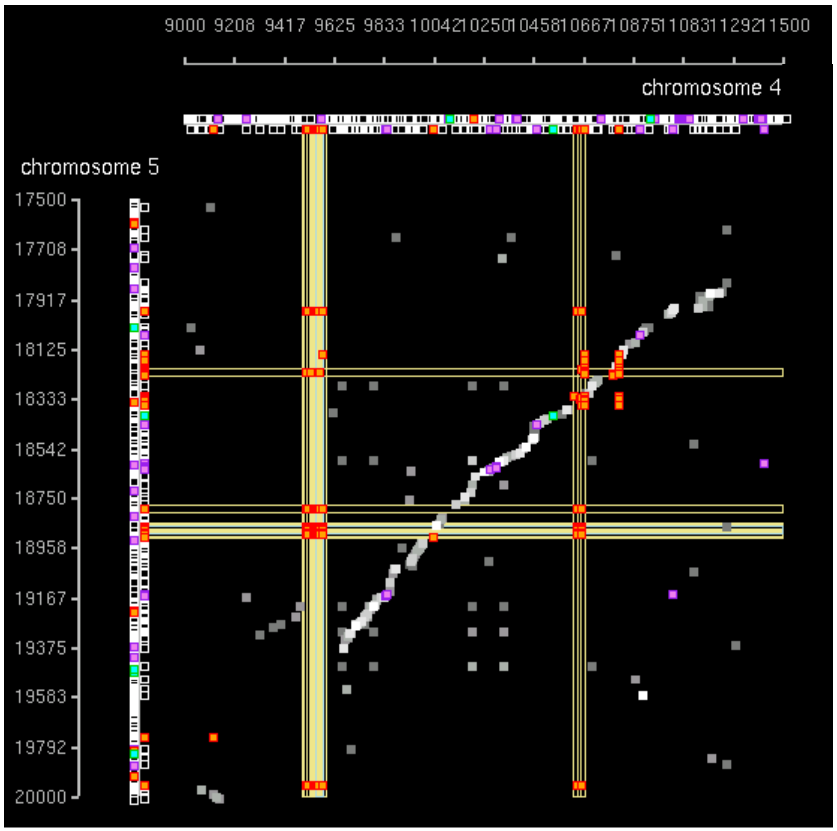
genome position — genome position

Visualization tools for studying ESTs, conserved orthologous sequences, and multigene families.
Alexander Kozik, UC Davis, Department of Vegetable Cropsv
http://www.atgc.org/GP_Ref/presentation/slide_14.html

Scalar to Scalar Mappings

circos

- if the domain in **g** and range in **g'** is small, a square dotter-like plot can be used

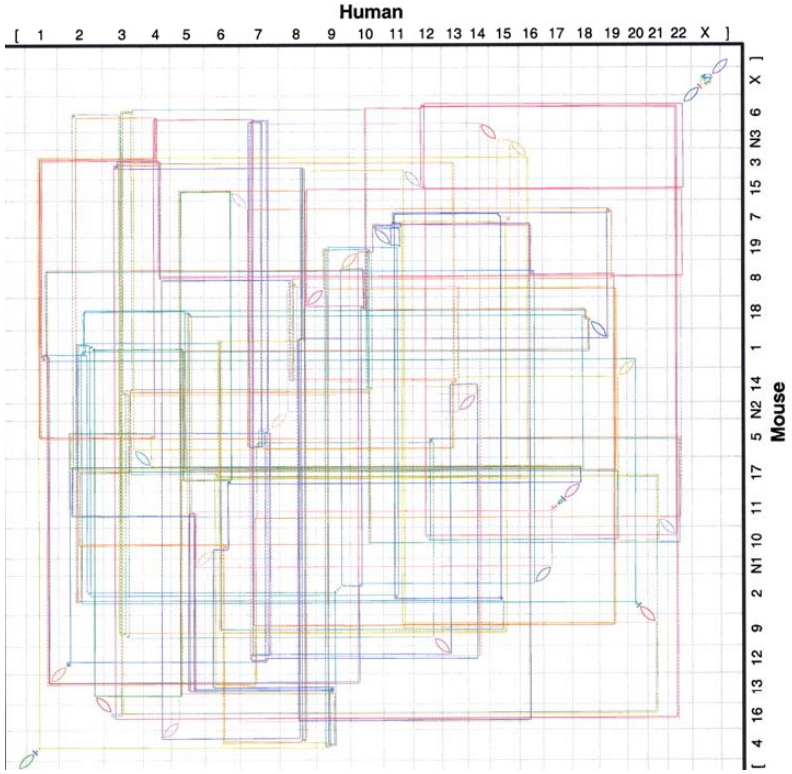


Visualization tools for studying ESTs,
conserved orthologous sequences, and multigene families.
Alexander Kozik, UC Davis, Department of Vegetable Cropsv
http://www.atgc.org/GP_Ref/presentation/slide_28.html

Genome-to-Genome Mappings

circos

- dotter-type plots in which **g** and **g'** are the entire genome, or span large distances, are hard to interpret
  - enormous dynamic range in data
  - routing lines becomes difficult

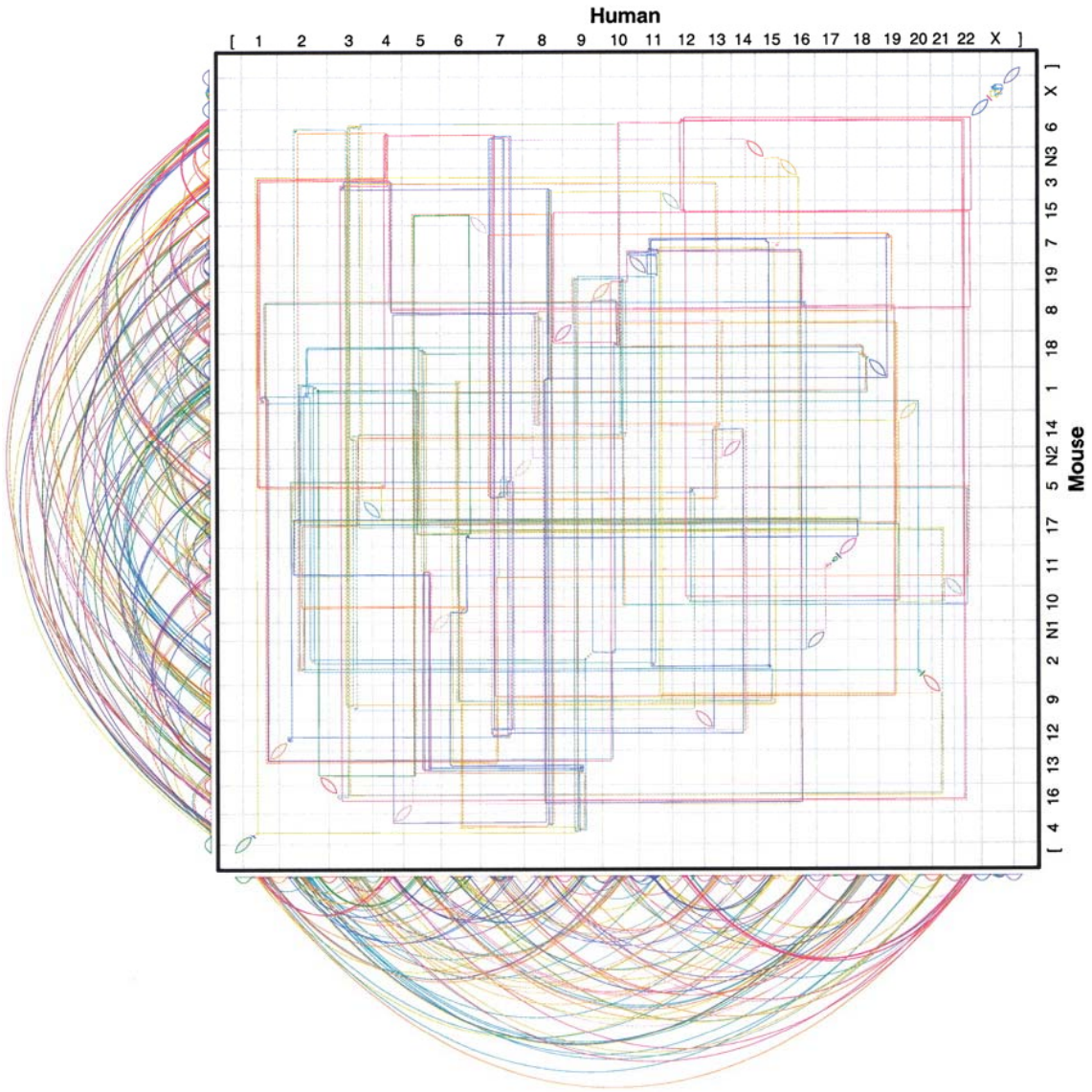


Genome Res. 2003 Jan;13(1):37-45

Genome-to-Genome Mappings

circos

- the problems in the standard 2-axis layout cannot be effectively mitigated
  - too much data
  - impossible to follow relationships within the data
- the figure hints at complexity
  - is the complexity introduced by the figure format?

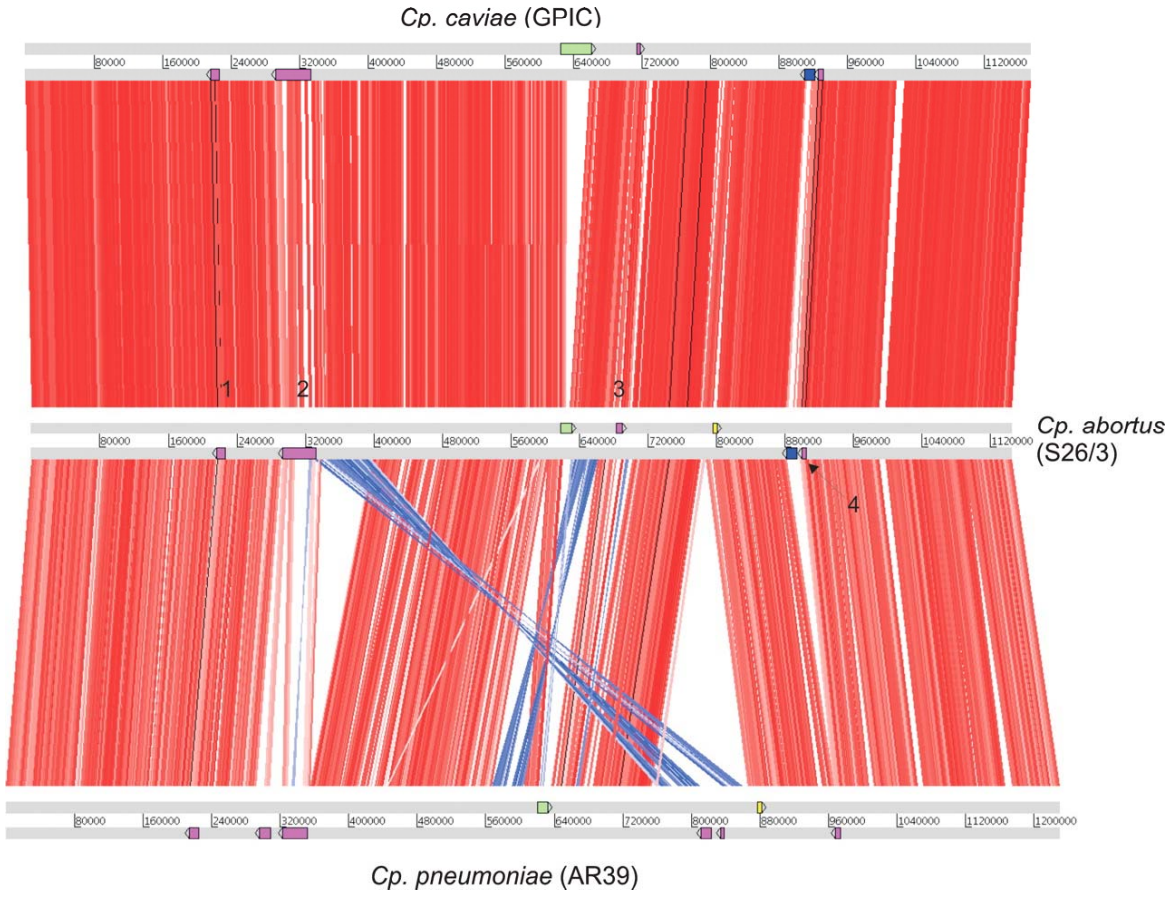


Genome Res. 2003 Jan;13(1):37-45

Genome-to-Genome Mappings

circos

- this is the most common way to represent relationships within genomic positions
  - works when the number of cross-overs is limited

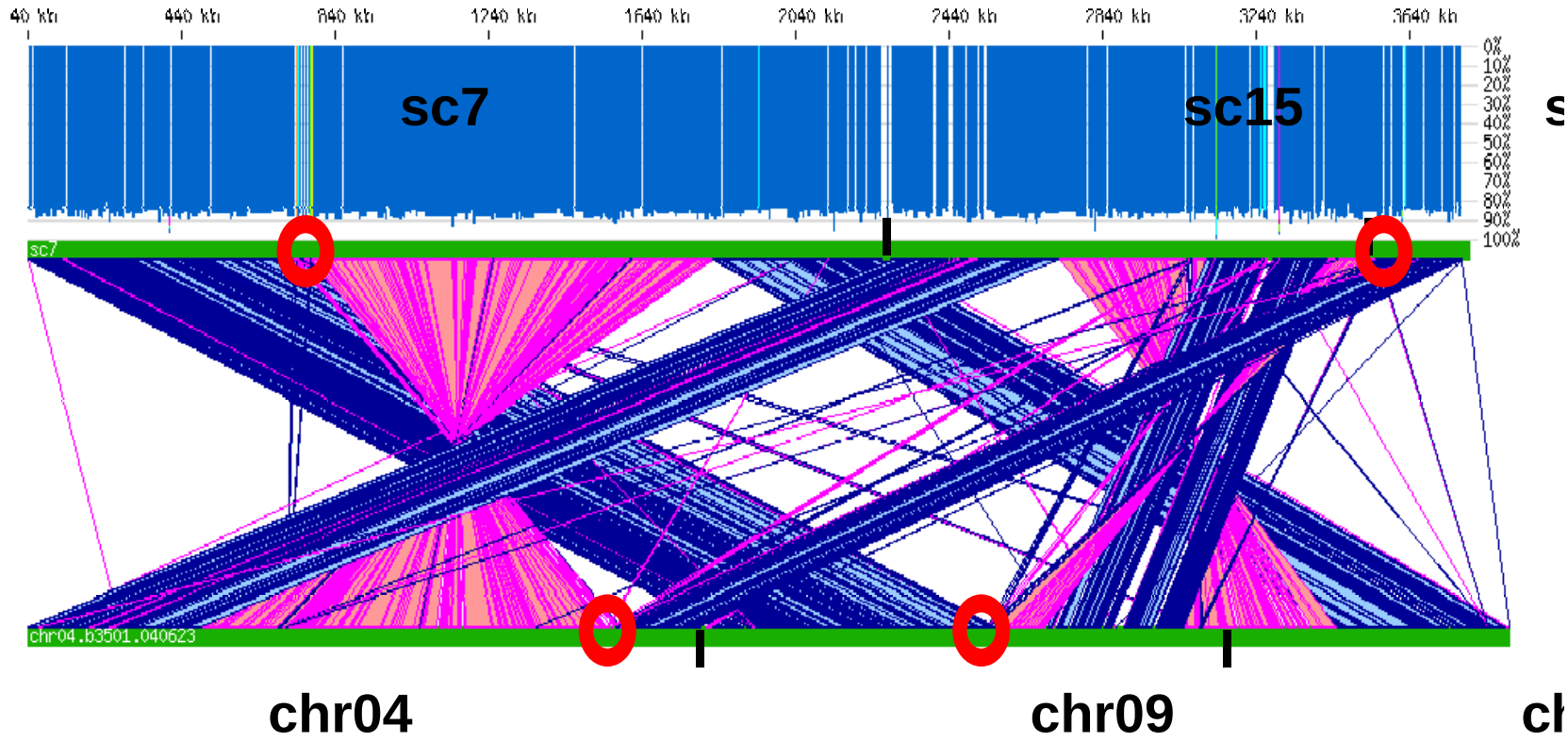


Genome **Res**. 2005 May;15(5):629-40

Genome-to-Genome Mappings

circos

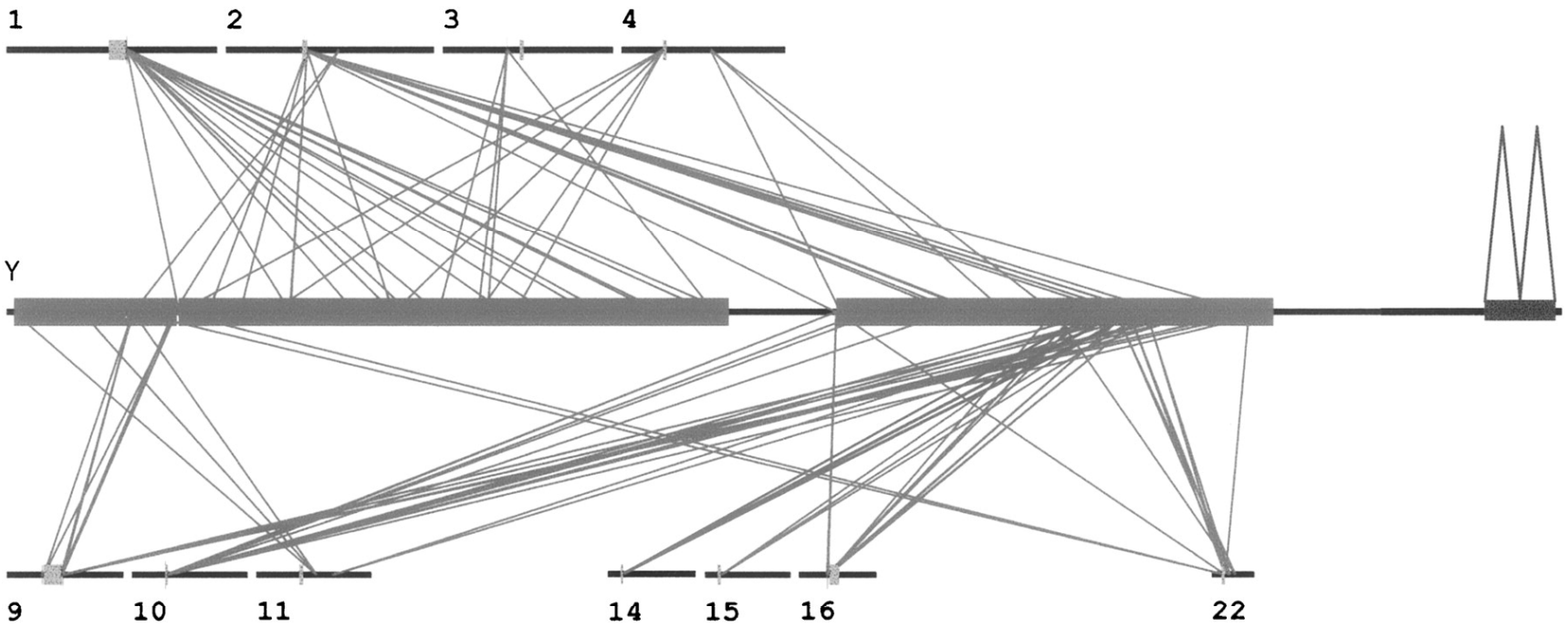
- works not so well when the number of cross-overs increases



Genome-to-Genome Mappings

circos

- when complexity is increased, the figure starts to lose cohesion
  - routing becomes difficult to follow
  - there is no focus point for the eye – your eye wanders over the figure



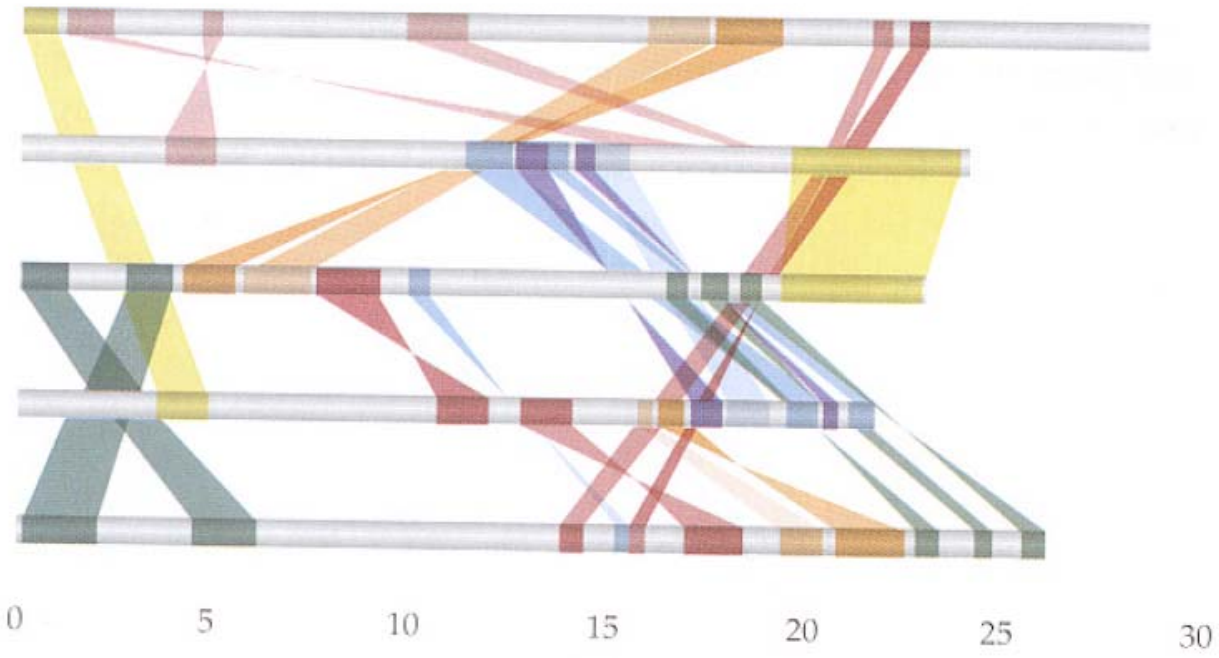
Genome Res. 2003 Jan;13(1):37-45

Genome-to-Genome Mappings

circos

- sometimes a little stylizing goes a long way
- custom images are time-consuming to create and difficult to automate

*Inter*chromosomal segmental duplications

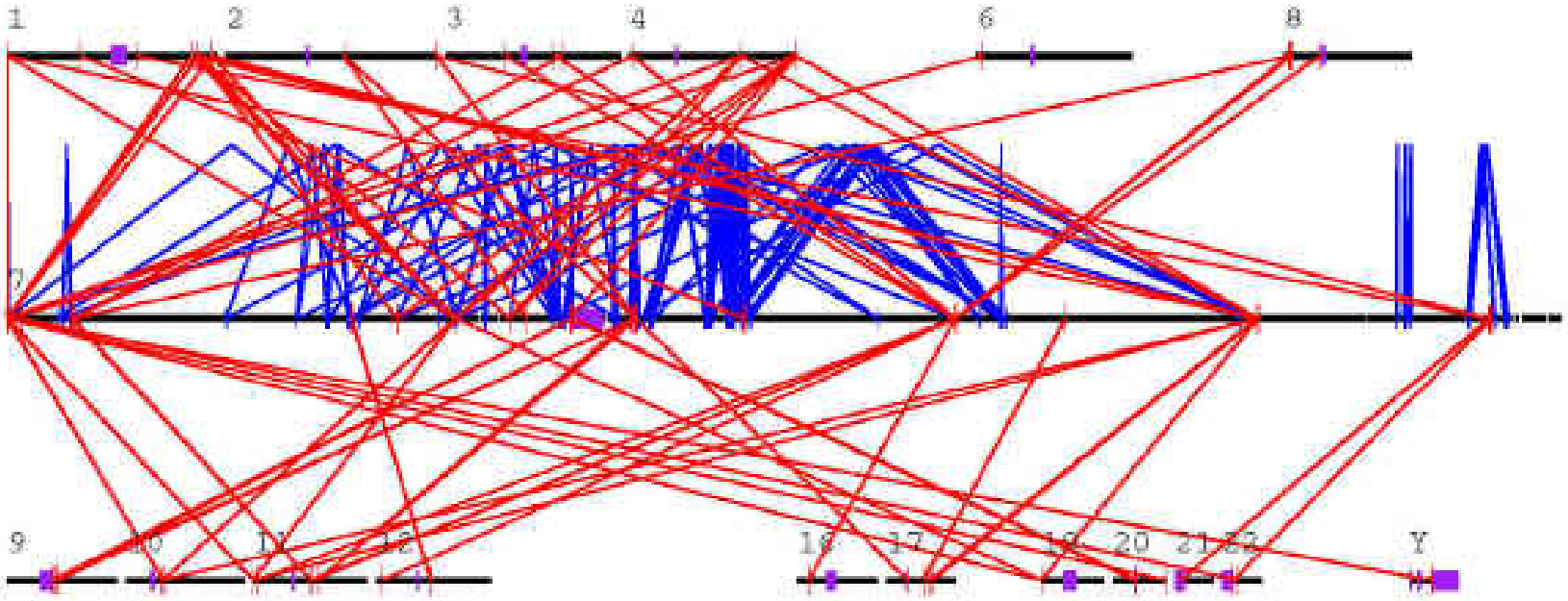


http://www.egg.isu.edu/Members/deborah/genomics

Genome-to-Genome Mappings

circos

- things get worse and worse when mappings that link both neighbouring (blue) and distant (red) positions are shown

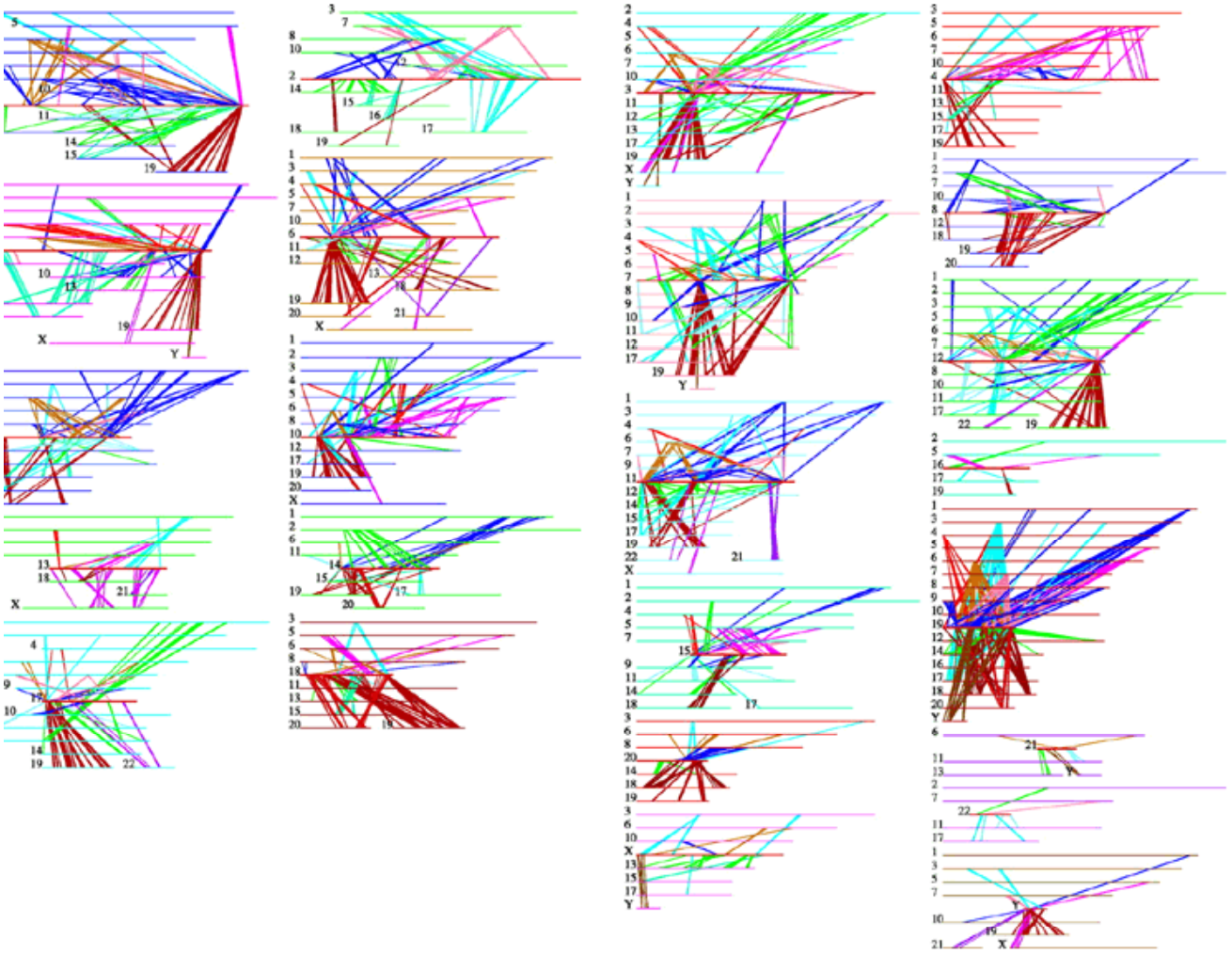


http://www.genome.wustl.edu/projects/human/chr7paper/chr7data/030113/segmental/index.php

Genome-to-Genome Mappings

circos

- you can try to fix things by partitioning your data set (somehow)
- mileage varies
  - generally poor

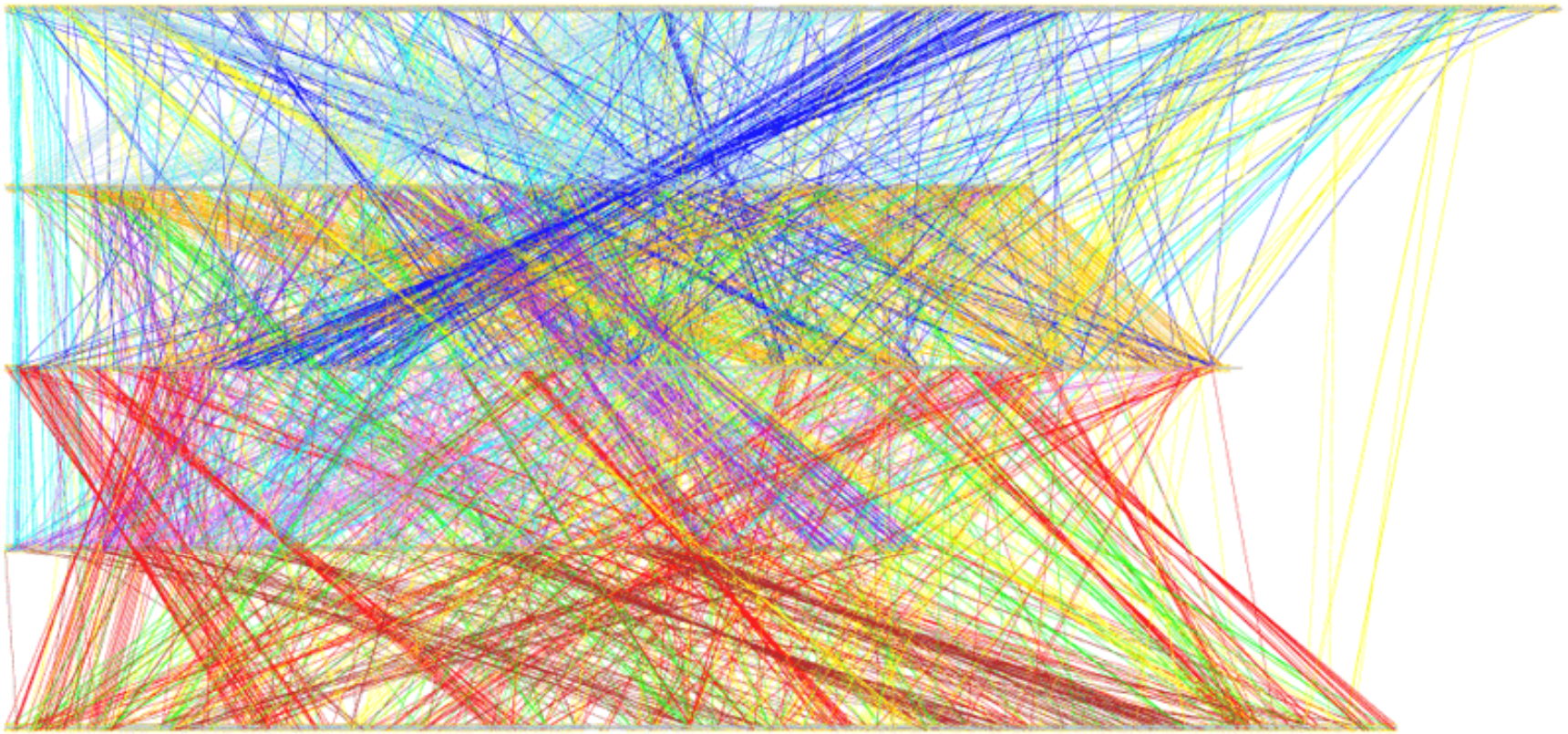


Venter, J. C., M. D. Adams, et al. (2001). "The sequence of the human genome." Science 291(5507): 1304-51.v

Genome-to-Genome Mappings

circos

- finally, you descend into data overload and information hell
  - this is not an informative plot, although a pretty one



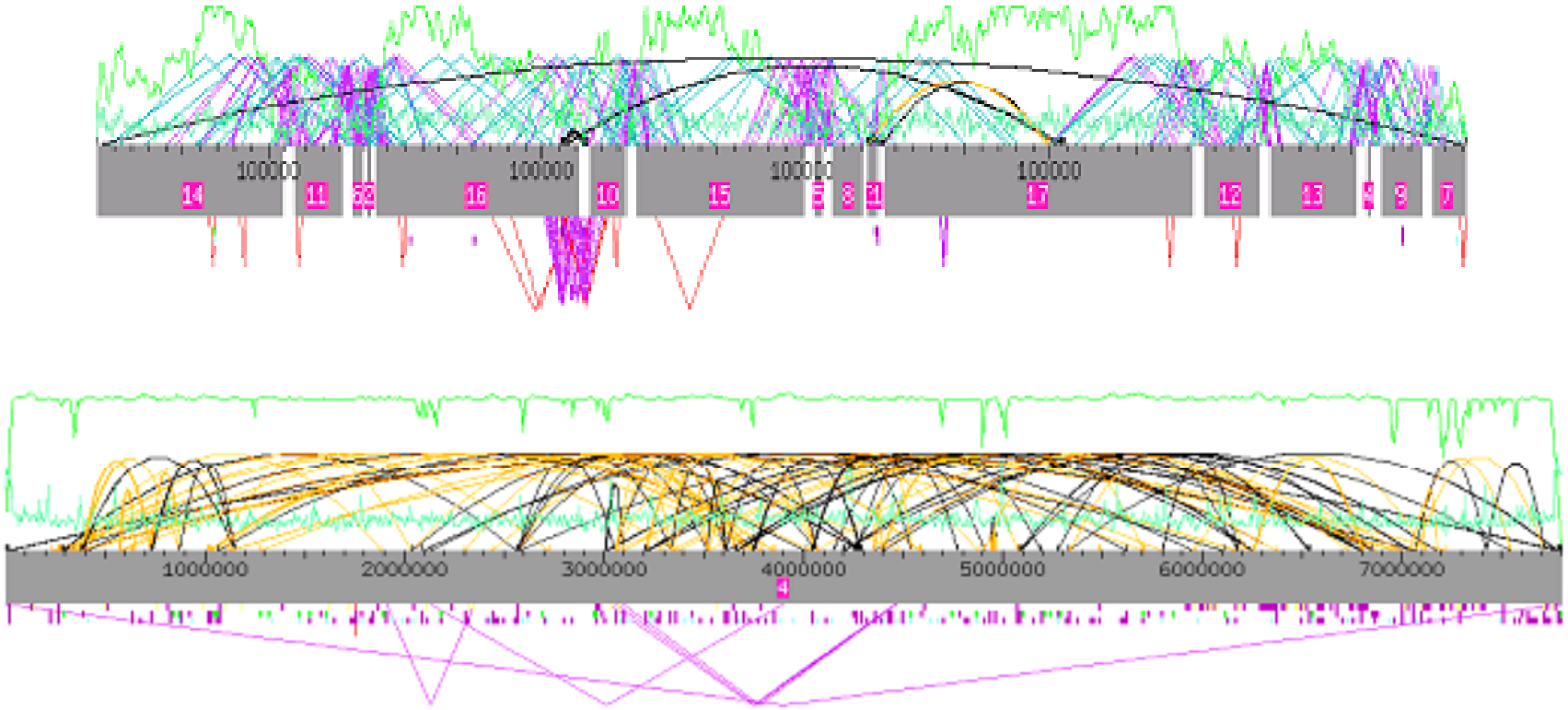
Segmental Duplications in Arabidopsis Genome. Alexander Kozik and Richard Michelmore, UC Davis, California

Image created with GenomePixelizer

Assembly Visualization

circos

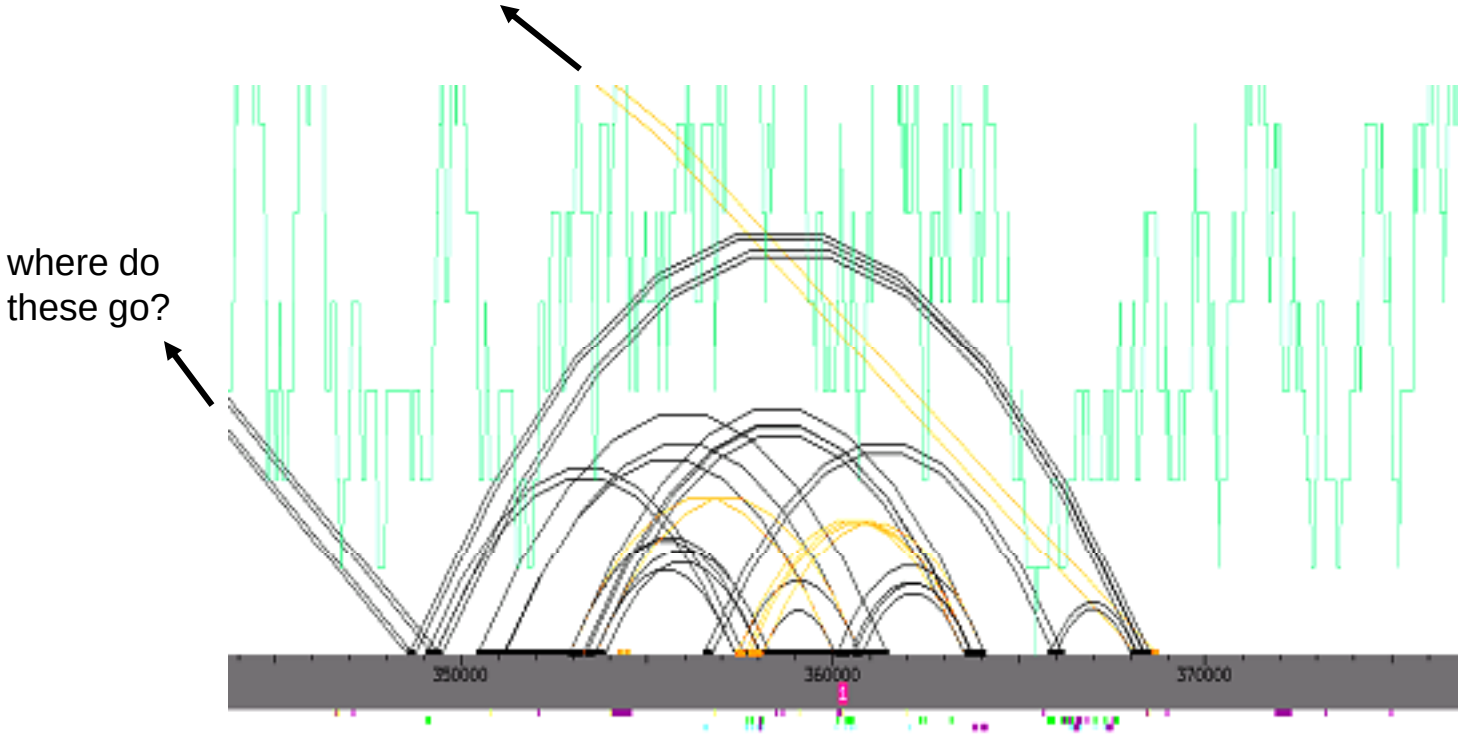
- Consed offers an assembly view
  - curves are nice, but too shallow when stretching across long distances
  - nice use of both sides of the axis



Assembly Visualization

circos

- zooming can provide more detail
  - but context is lost



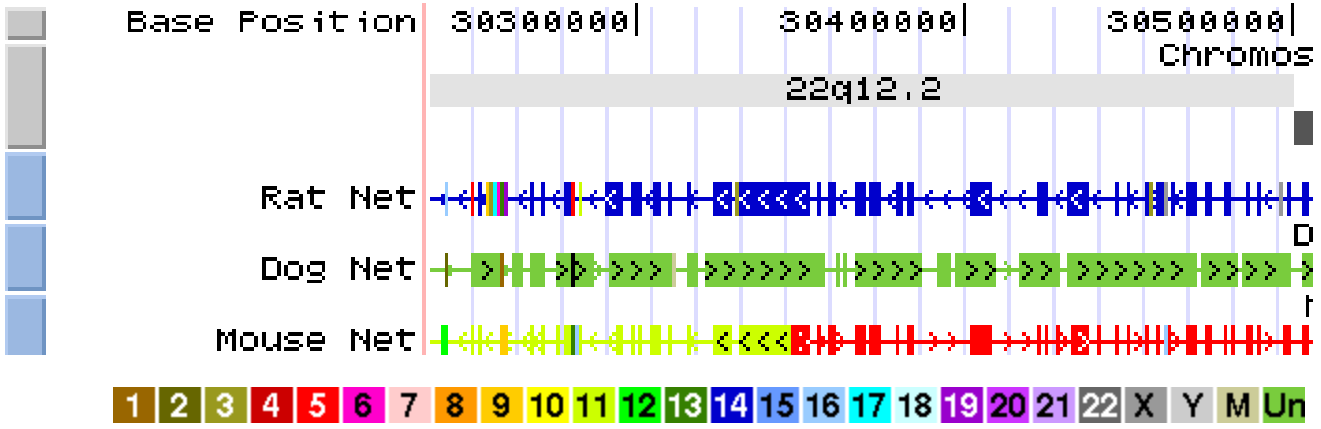
What Do We Do?

circos

- work with smaller genomes
  - I wish!

- reduce information content in figures
  - distill target genome position to a colour, based on target chromosome

$$f : g \rightarrow g' \rightarrow c'$$



UCSC Genome Browser (hg17)

Reducing Information Content

circos

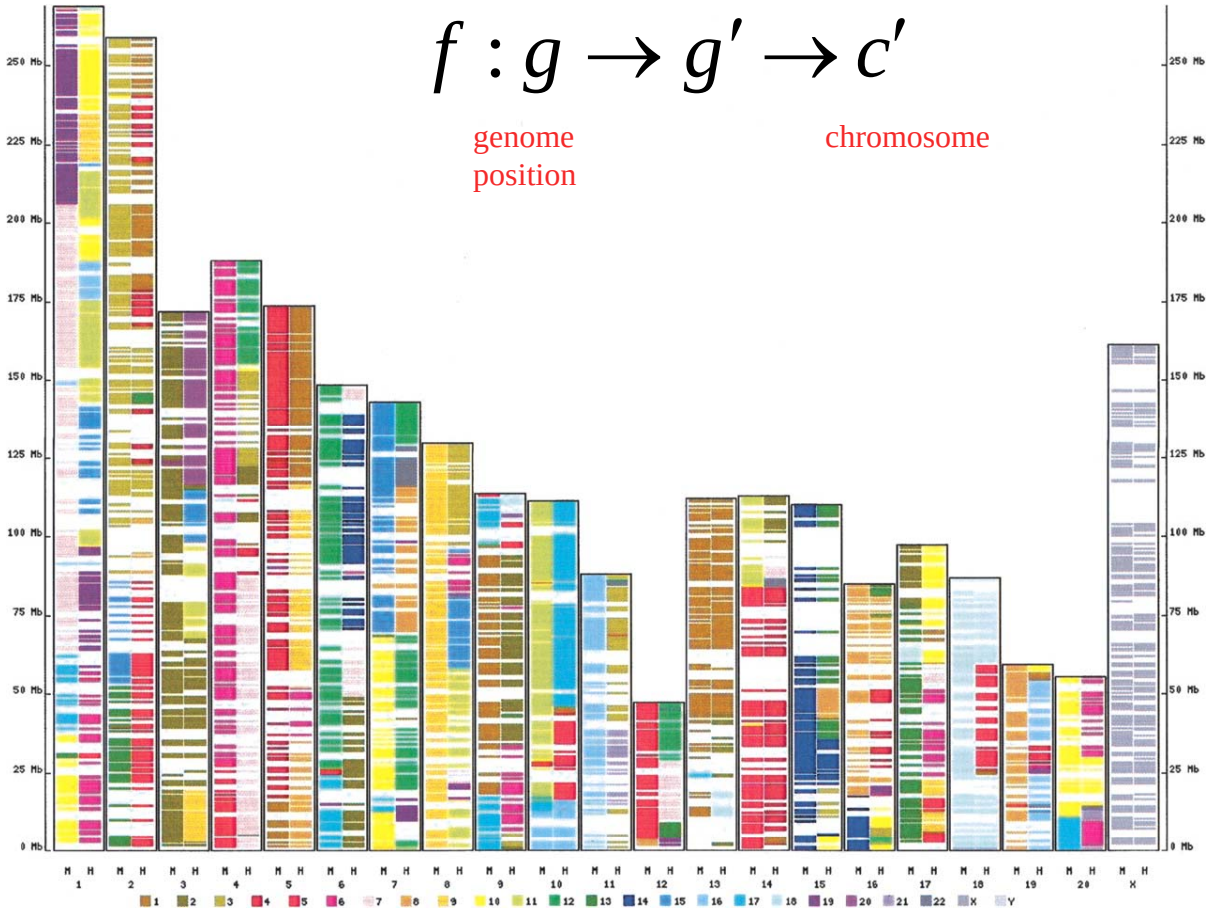
- draw the domain, colour regions in the domain by reduced representation of range
  - target chromosome, by colour

$$f : g \rightarrow g' \rightarrow c'$$

genome position

chromosome

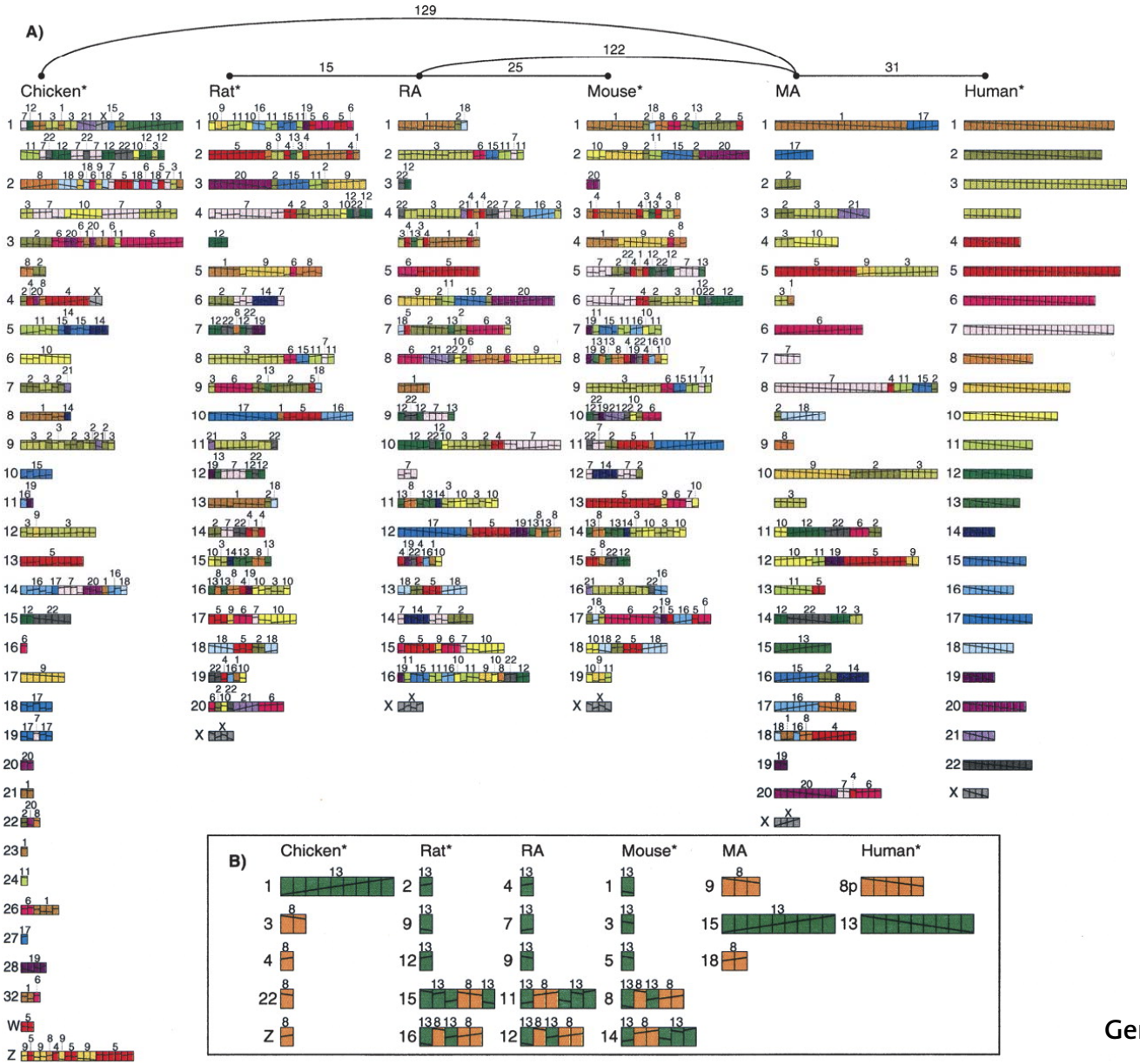
colour scheme convention



Genome Res. 2004 Apr;14(4):685-92

Reducing Information Content

circos

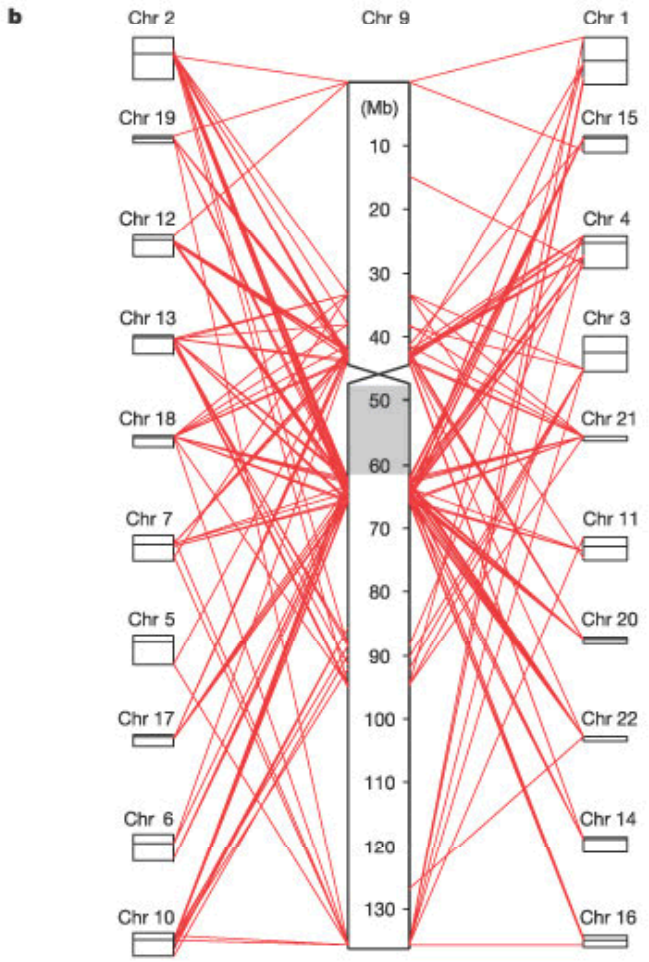


Genome Res. 2005 Jan;15(1):98-110

Alter Information Layout

circos

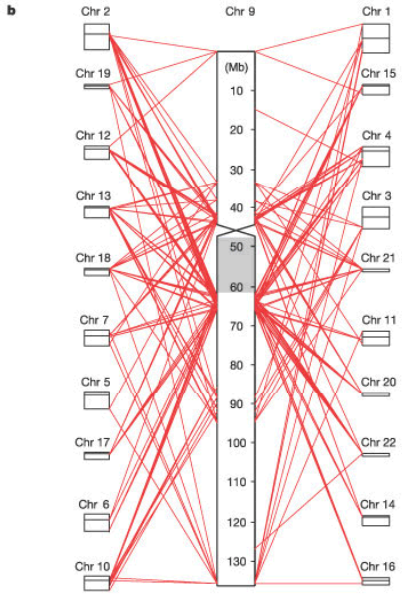
- altering axes layout can help
  - reduce cross-overs
  - draw focus to regions of interest
    - source/sink of lines
    - deserts
- however, note how the order of the peripheral chromosomes in this figure is unconventional



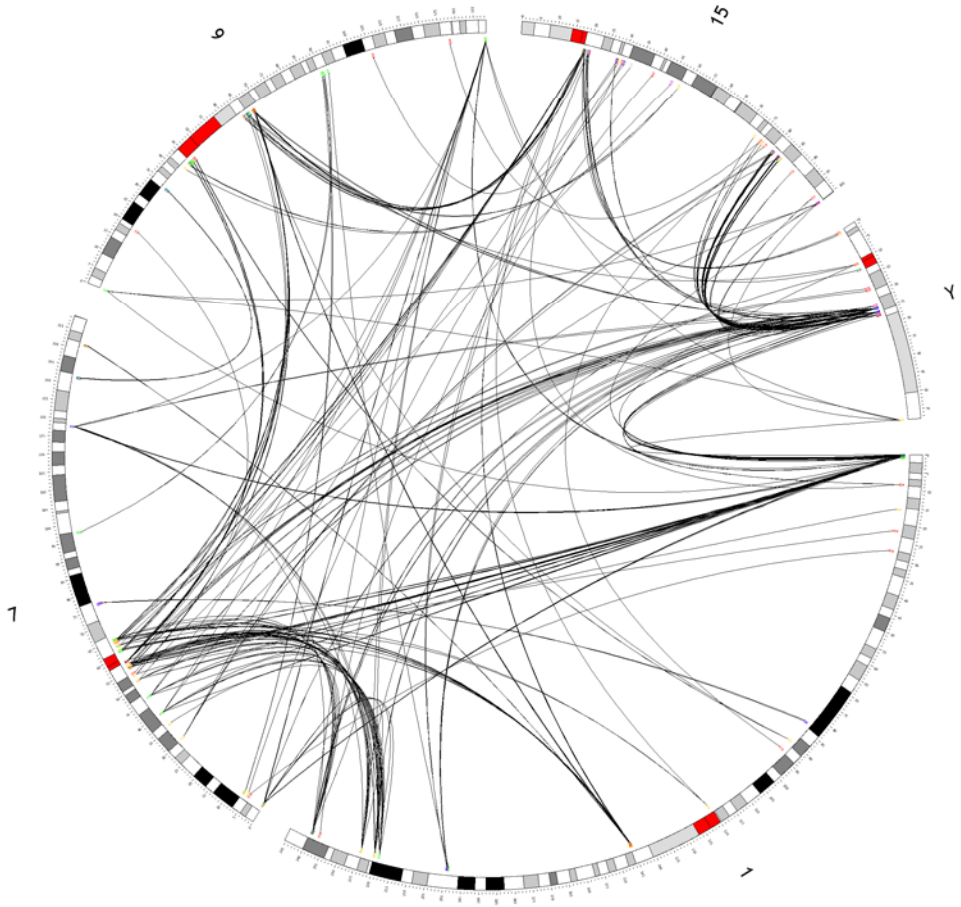
Humphray, S. J., K. Oliver, et al. (2004).
"DNA sequence and analysis of human chromosome 9."
Nature 429(6990): 369-74.

Alter Information Layout

circos



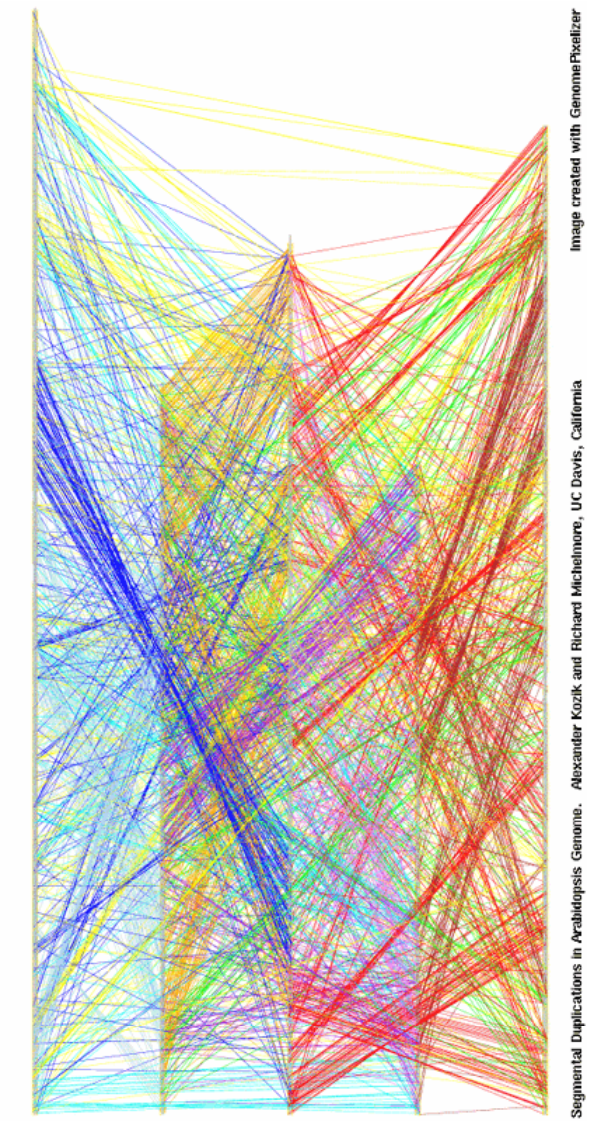
Humphray, S. J., K. Oliver, et al. (2004).
"DNA sequence and analysis of human chromosome 9."
Nature 429(6990): 369-74.

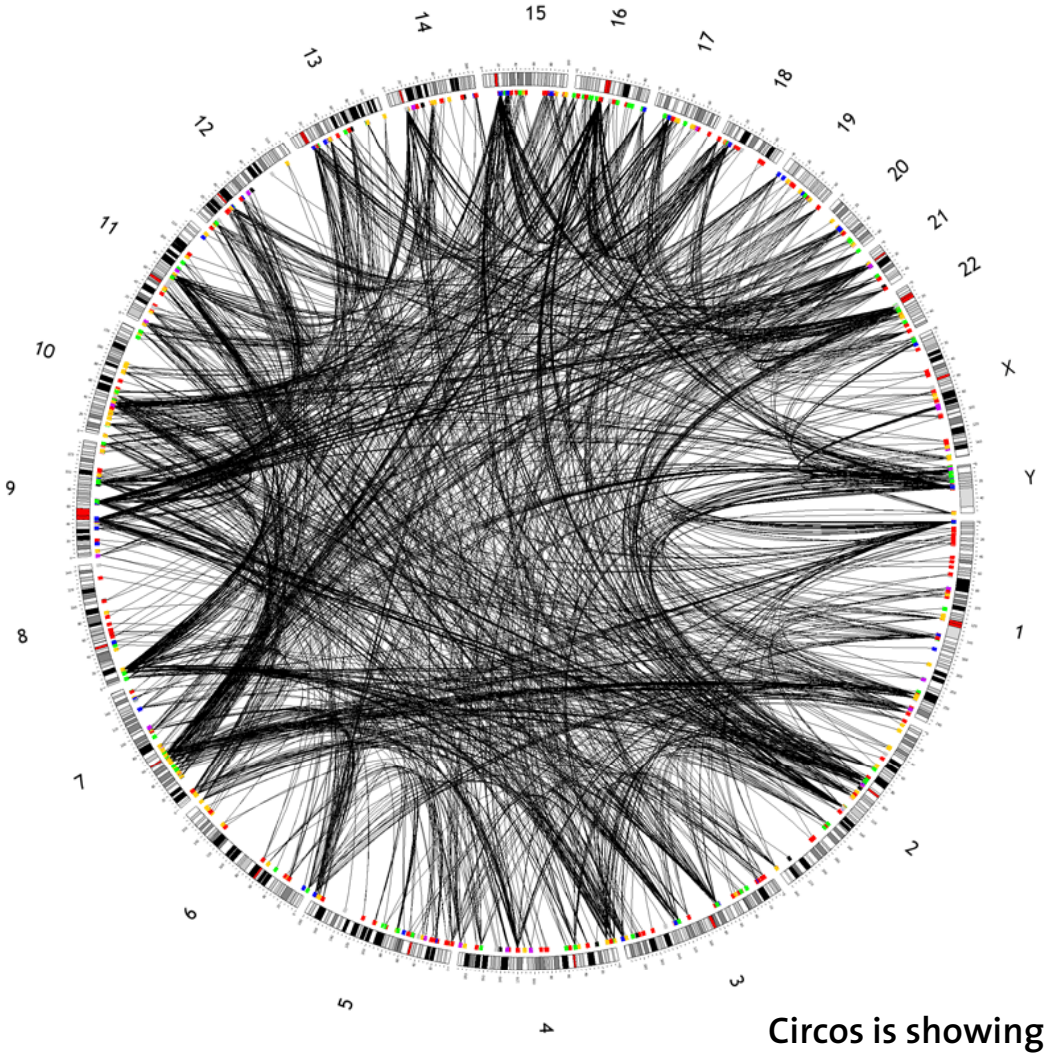


Circos image

Alter Information Layout

circos

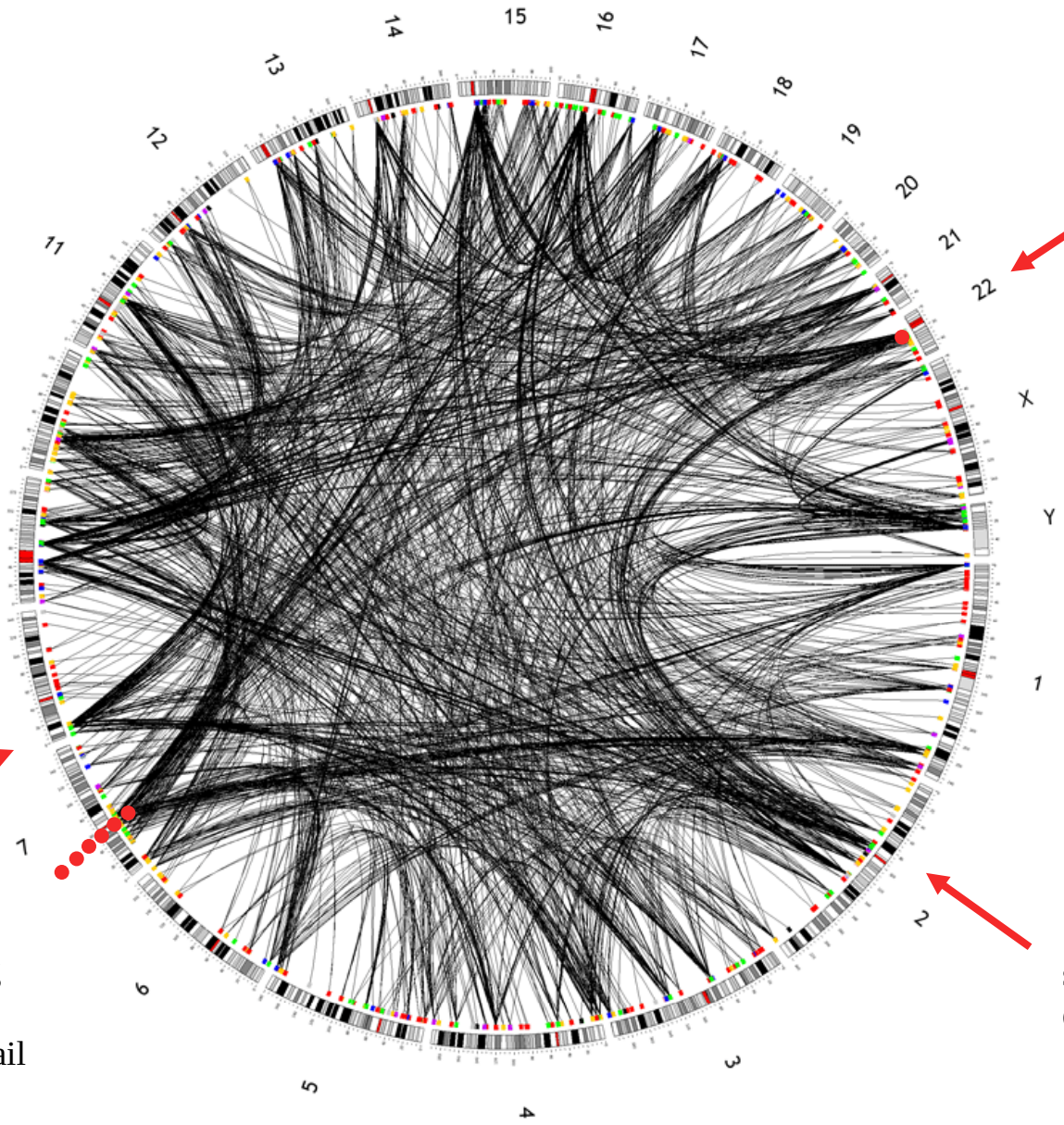




Circos is showing 22,000 lines

Benefits of circular composition

circos



sinks/sources
easy to see

sinks/sources
easy to see

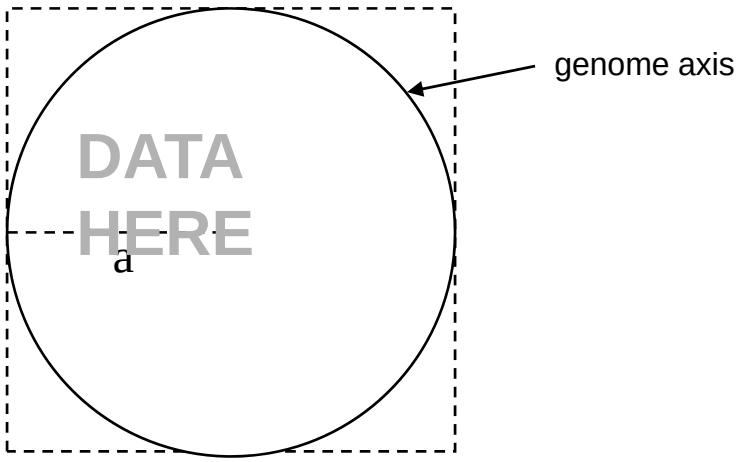
interior lines
make routing
easy, while
retaining detail

sinks/sources
easy to see

Winner: Circle

circos

- the circle is more symmetric than square – eye is less burdened
- circle's data payload is higher
  - consider the ratio of the **axis length** to the **data area**
    - for a square: $2a/4a^2 = 1/2a$ (2a = sum of x,y axes lengths)
    - for a circle: $2\pi a/\pi a^2 = 2/a$ (4 times larger)
- concentric tracks are more efficient
  - (+) more efficient use of figure area – longer axis allows for greater spatial detail
  - (-) $\Delta r \Delta \varphi$ is not constant in area ($\Delta x \Delta y$ is) – shape is distorted



Circos

circos

- Perl
- graphics by GD (API to gd graphics library)
- Apache-like configuration file

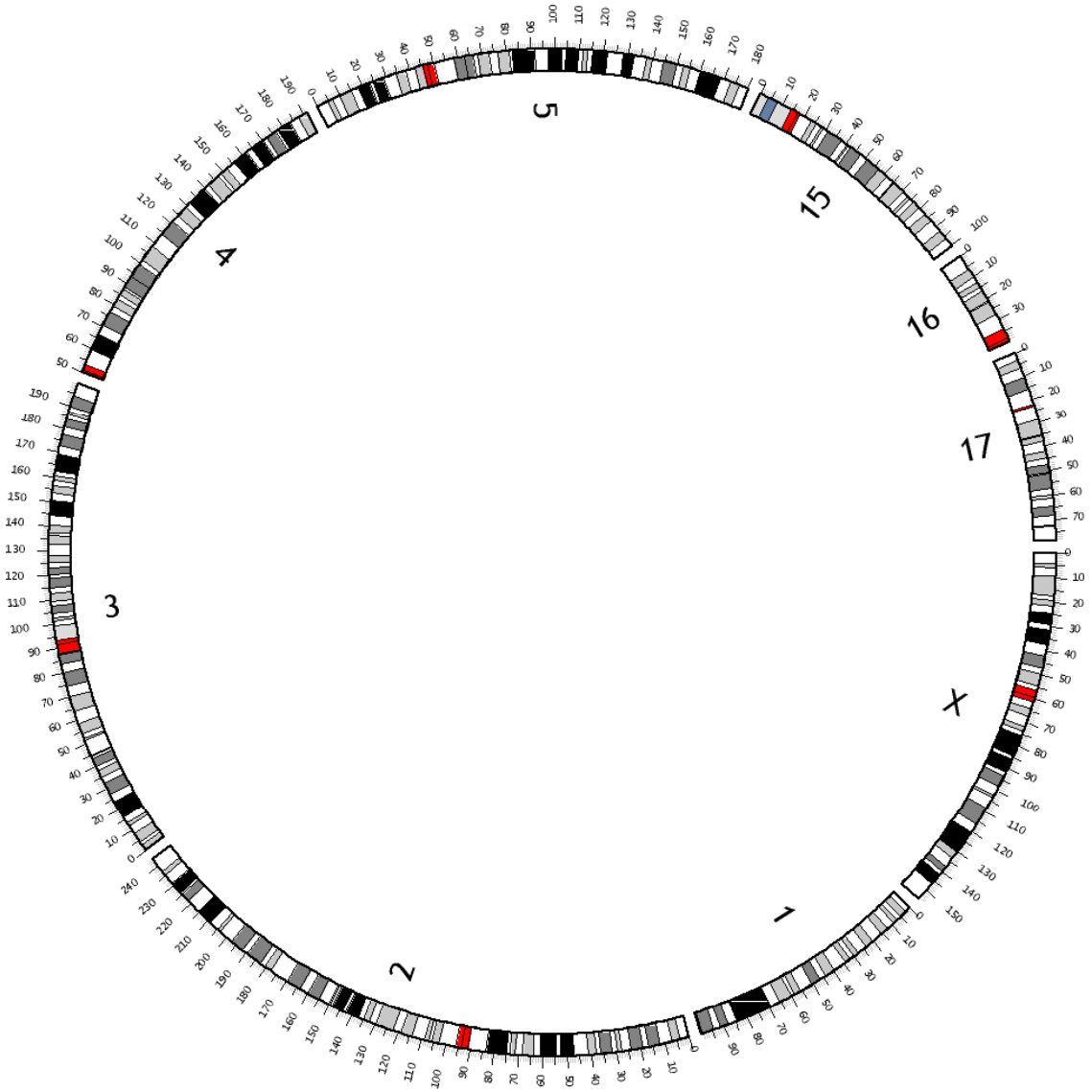
- *mkweb.bcgsc.ca/circos*

- features
  - generalized concentric data tracks
    - line, scatter, histogram
    - clone tiles
    - mappings
      - dynamic geometry/line property rules
  - non-linear scale
    - regions can be locally zoomed without cropping
  - full user control over aspects of all elements
    - colour, thickness, stroke, etc

Circular Axis

circos

- start with objects that have a distance scale
  - chromosome
  - contig
  - sequence
  - map
- place objects around the circle
  - order can be optimized for better routing
- superimpose data tracks



Configuration File

circos

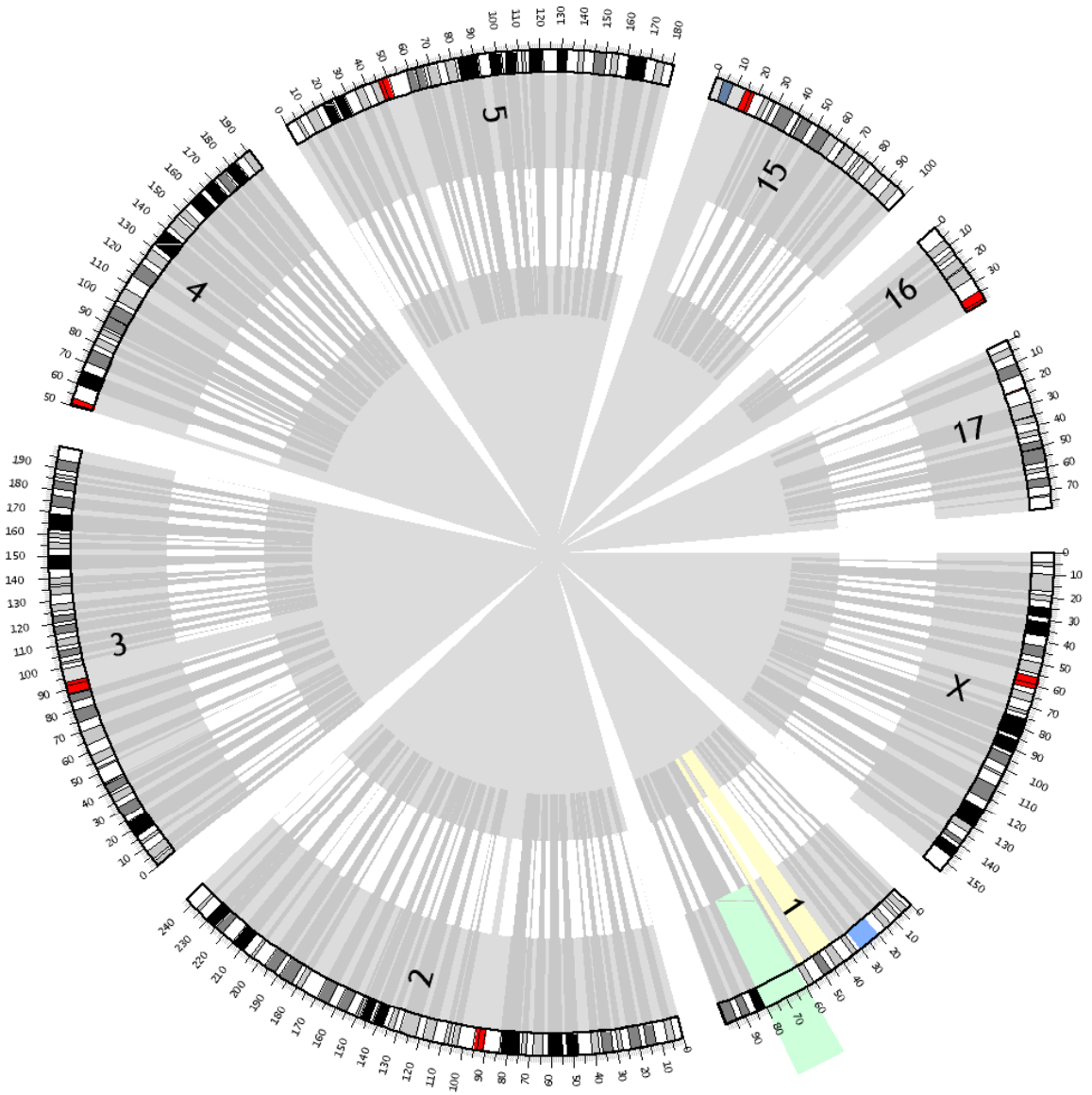
```
<colors>
<<include ../etc/colors.conf>>
</colors>

karyotype   = ../data/karyotype_hg17.txt

outputdir   = /home/martink/www/htdocs/circos/tutorial/001
outputfile  = 4.gif

radius       = 500

chrspacing     = 5e6
chrthickness   = 20
chrstroke      = 2
chrcolor       = black

chrradius      = 0.9
chrlabel       = yes
chrlabelradius = 0.75
chrlabelsize   = 24

bandstroke      = 1
showbands       = yes
fillbands       = yes

chromosomes = 1:0-100000000,2,3,4:50000000-,5,15,16:-40000000,17,X

chrticklabels     = yes
tickmultiplier    = 1e-6
tickradiusoffset  = 0.0
gridoffset        = 0
gridstart         = 0.55
```

```
<ticks>
<tick>
spacing   = 1000000
size      = 5
thickness = 1
color     = grey
label     = no
labelsize = 12
format    = %d
grid      = no
</tick>
<tick>
spacing   = 5000000
size      = 7
thickness = 1
color     = black
label     = no
labelsize = 6
format    = %.1f
grid      = no
gridcolor = grey
</tick>
<tick>
spacing   = 10000000
size      = 10
thickness = 1
color     = black
label     = yes
labelsize = 8
format    = %d
grid      = no
gridcolor = dgrey
</tick>
</ticks>
```

Highlights

circos

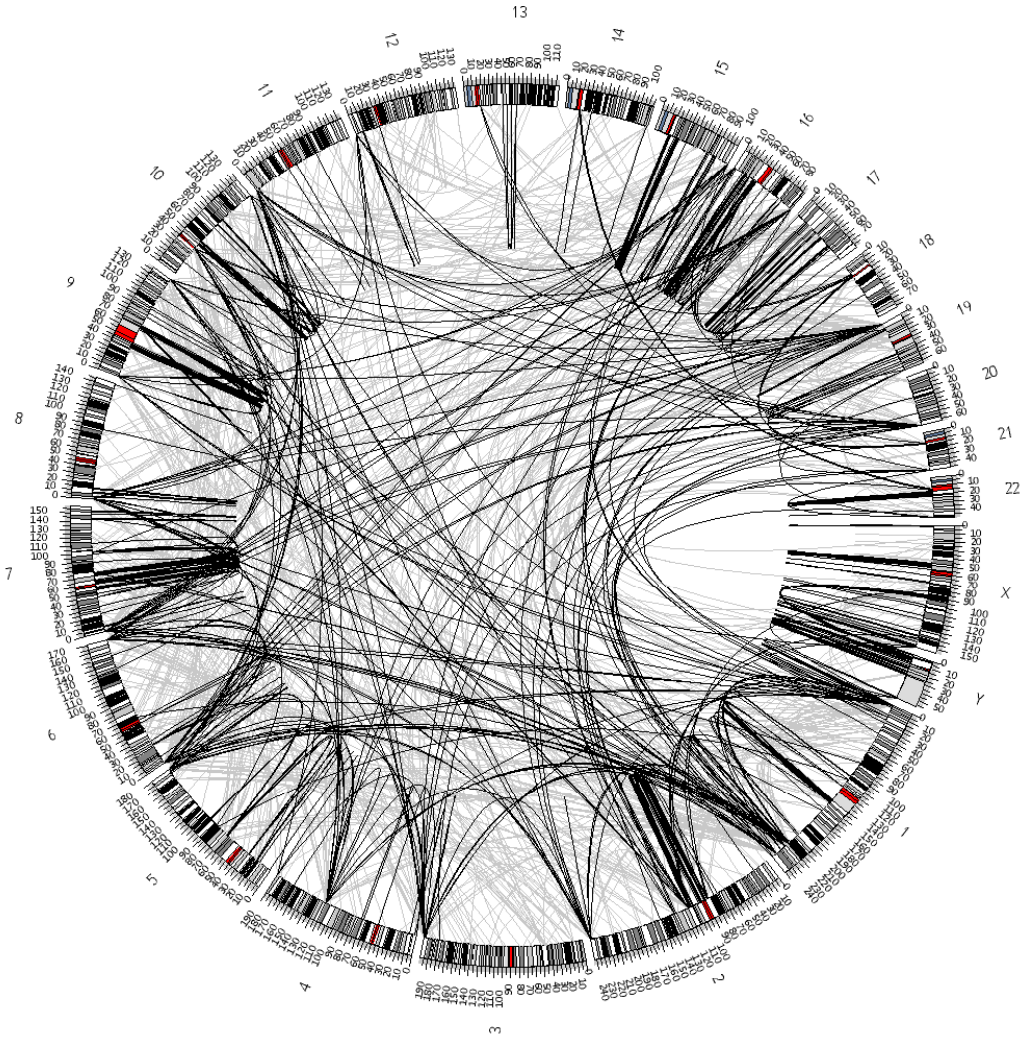
- you can **highlight** regions by creating coloured slices
  - order of layering controlled by z-level for each element
  - highlights sit in the back, under all other elements



Genome-to-Genome Mappings

circos

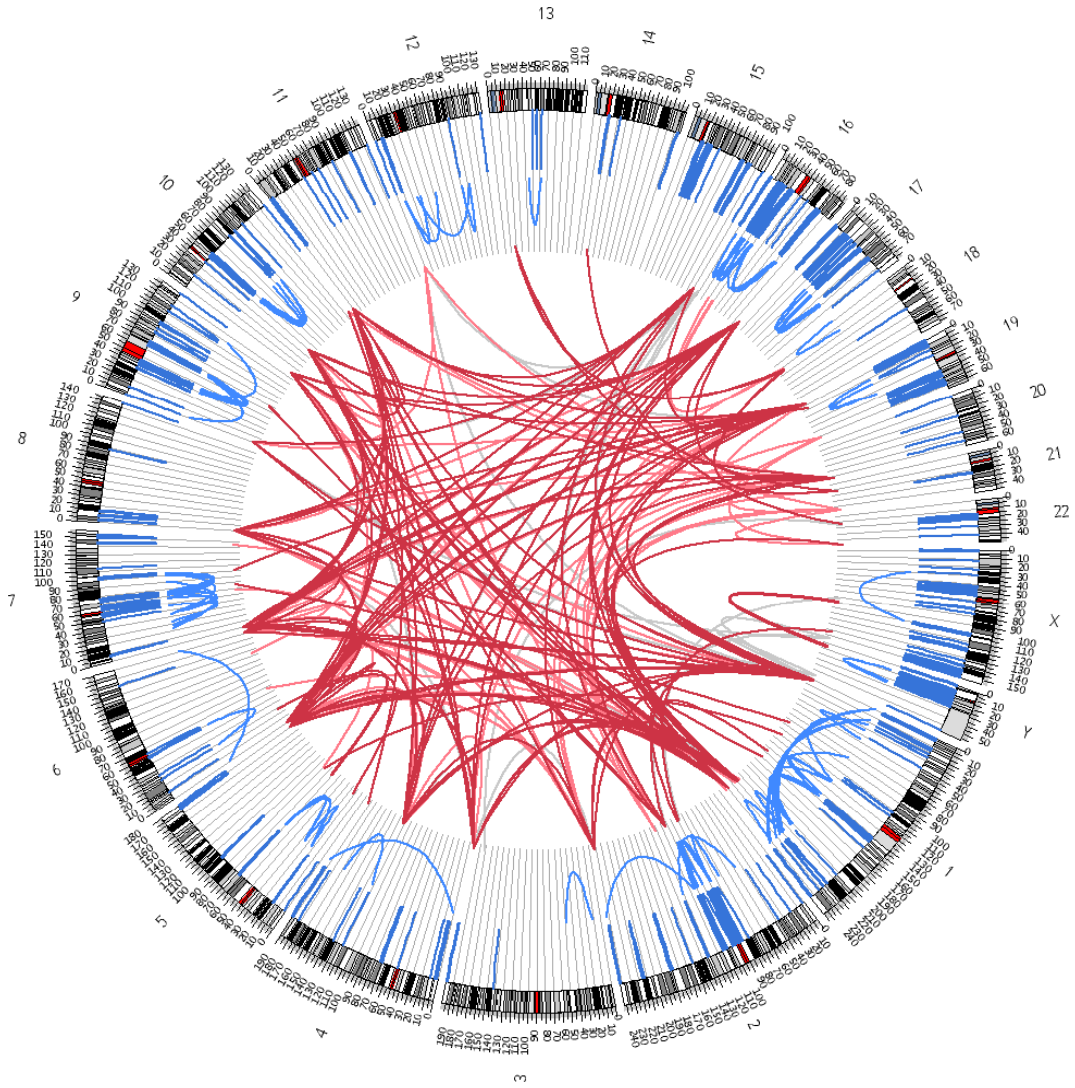
```
# in configuration file

<links segdup>
  show         = yes
  color        = black
  thickness    = 1
  offset       = 0
  bezierradius = 0.3
  file         = segdups.txt
</links>
```



```
# segdups.txt format
# ID chr1 pos11 pos12
# ID chr2 pos21 pos22
. . .
segdup10133 13 17975618 17981753
segdup10133 4 131149507 131155638
segdup10148 4 131149510 131152617
segdup10148 4 131156685 131159786
segdup10156 1 143389520 143392018
segdup10156 4 131156687 131159175
segdup10161 13 17989958 17991102
segdup10161 4 131158639 131159786
. . .
```

Formatting Rules

```
<links segdup98>
  show         = yes
  color        = grey
  thickness    = 2
  offset       = 0
  bezierradius = 0.2
  file         = segdups.txt
  z            = 0

  <rule link>
  FORMATTING RULE
  </rule>
  . . .
  <rule link>
  FORMATTING RULE
  </rule>

</links>
```



Formatting Rules

circos

1
```
rule  = '_CHR1_' eq '_CHR2_' && abs(_POS1_-_POS2_) < 10000000
color = blue
bezierradius = 0.7
```

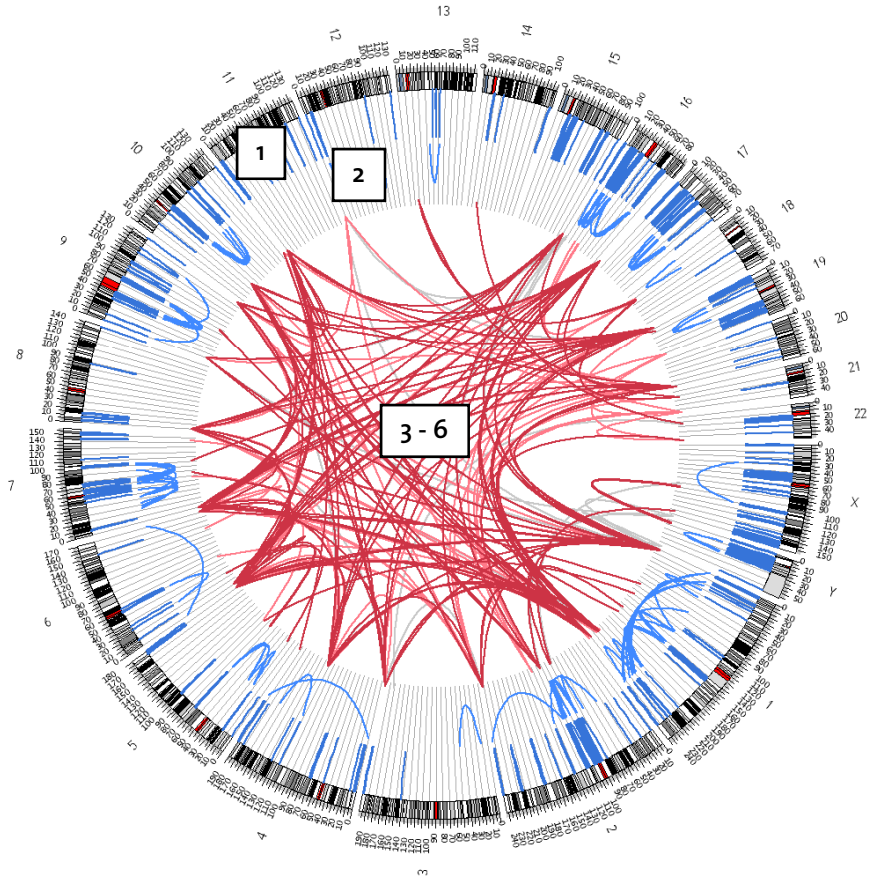
2
```
rule   = '_CHR1_' eq '_CHR2_' && abs(_POS1_-_POS2_) >= 10000000
color  = lblue
offset = 0.125
bezierradius = 0.6
```

3
```
rule   = '_CHR1_' ne '_CHR2_' && min(_SIZE1_,_SIZE2_) >= 25
offset = 0.25
color  = dred
z = 10
importance = 20
```

4
```
rule   = '_CHR1_' ne '_CHR2_' && min(_SIZE1_,_SIZE2_) > 100
offset = 0.25
color  = lred
z = 7
importance = 10
```

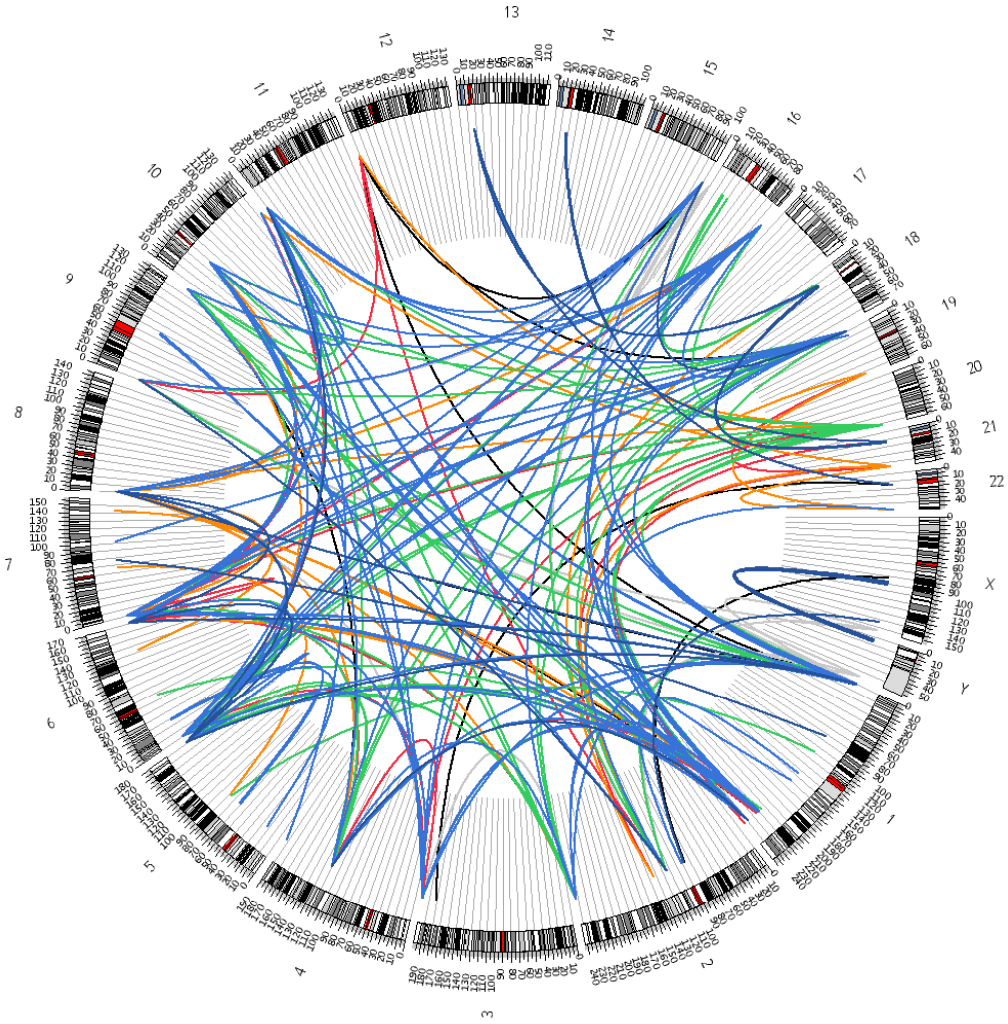
5
```
rule   = '_CHR1_' ne '_CHR2_' && min(_SIZE1_,_SIZE2_) > 500
offset = 0.25
color  = grey
importance = 5
z = 5
```

6
```
rule   = '_CHR1_' ne '_CHR2_'
offset = 0.25
color  = vlred
z = 5
hide = yes
```



Formatting Rules

```
<rule link>
importance = 100
rule  = '_CHR1_' eq '_CHR2_'
hide = yes
</rule>

<rule link>
importance = 100
rule   = '_CHR1_' ne '_CHR2_' && min(_SIZE1_,_SIZE2_) < 5000
hide = yes
</rule>

<rule link>
importance = 90
rule   = '_CHR1_' ne '_CHR2_' && min(_SIZE1_,_SIZE2_) < 7500
color  = black
z = 0
</rule>

<rule link>
importance = 85
rule   = '_CHR1_' ne '_CHR2_' && min(_SIZE1_,_SIZE2_) < 10000
color  = grey
z = 5
</rule>

<rule link>
importance = 80
rule   = '_CHR1_' ne '_CHR2_' && min(_SIZE1_,_SIZE2_) < 15000
color  = red
z = 10
</rule>

<rule link>
importance = 75
rule   = '_CHR1_' ne '_CHR2_' && min(_SIZE1_,_SIZE2_) < 20000
color  = orange
z = 15
</rule>

. . .
```



Formatting Rules

circos

1

```
<rule link>
importance   = 100
rule         = '_CHR1_' eq '_CHR2_'
               && abs(_POS1_-_POS2_) < 20000000
bezierradius = 0.8
crest        = 0.1
color        = grey
offset       = 0
z            = -10
</rule>
```

2

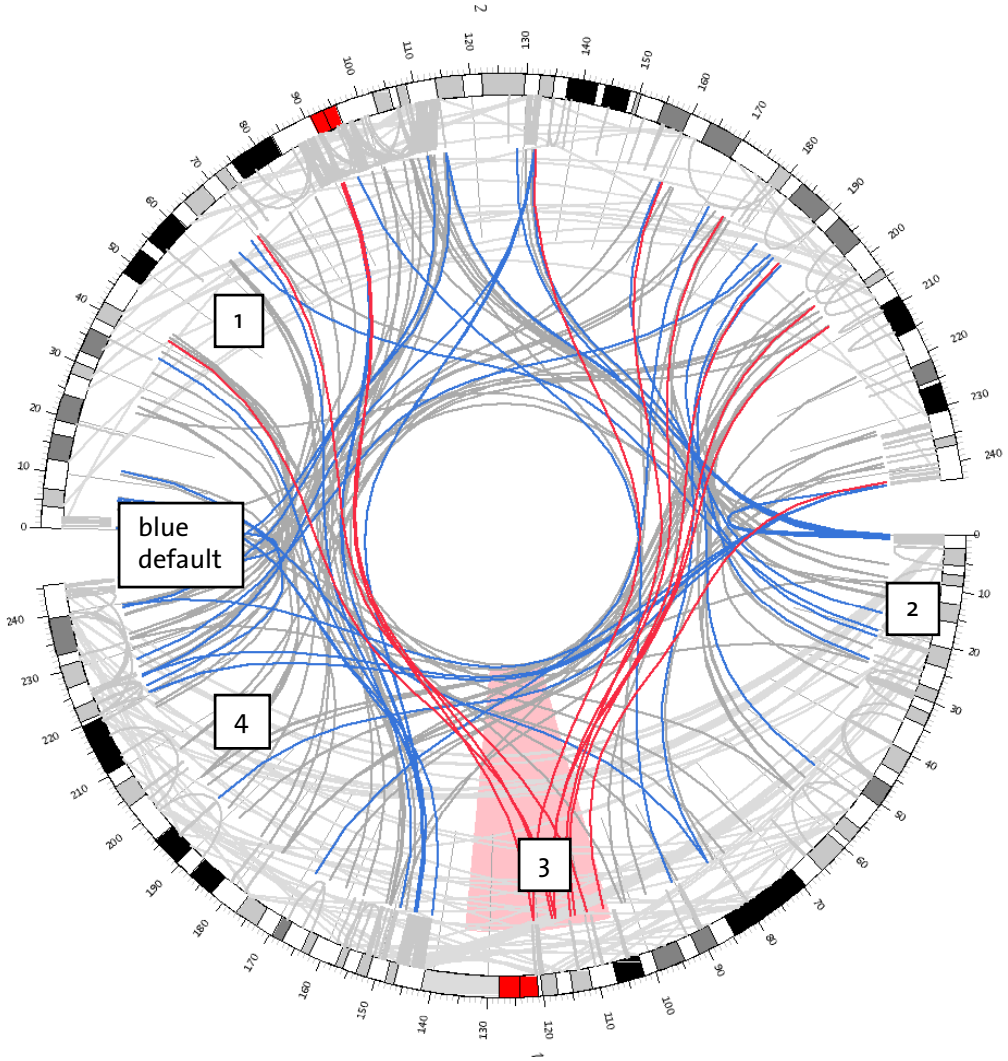
```
<rule link>
importance   = 100
rule         = '_CHR1_' eq '_CHR2_'
               && abs(_POS1_-_POS2_) >= 20000000
bezierradius = 0.9
crest        = 0
color        = lgrey
offset       = 0
z            = -20
</rule>
```

3

```
<rule link>
importance   = 90
rule         = _CHR1_ eq "1"
               && abs(_POS1_ - 120000000) < 15000000
color        = red
z            = 15
</rule>
```

4

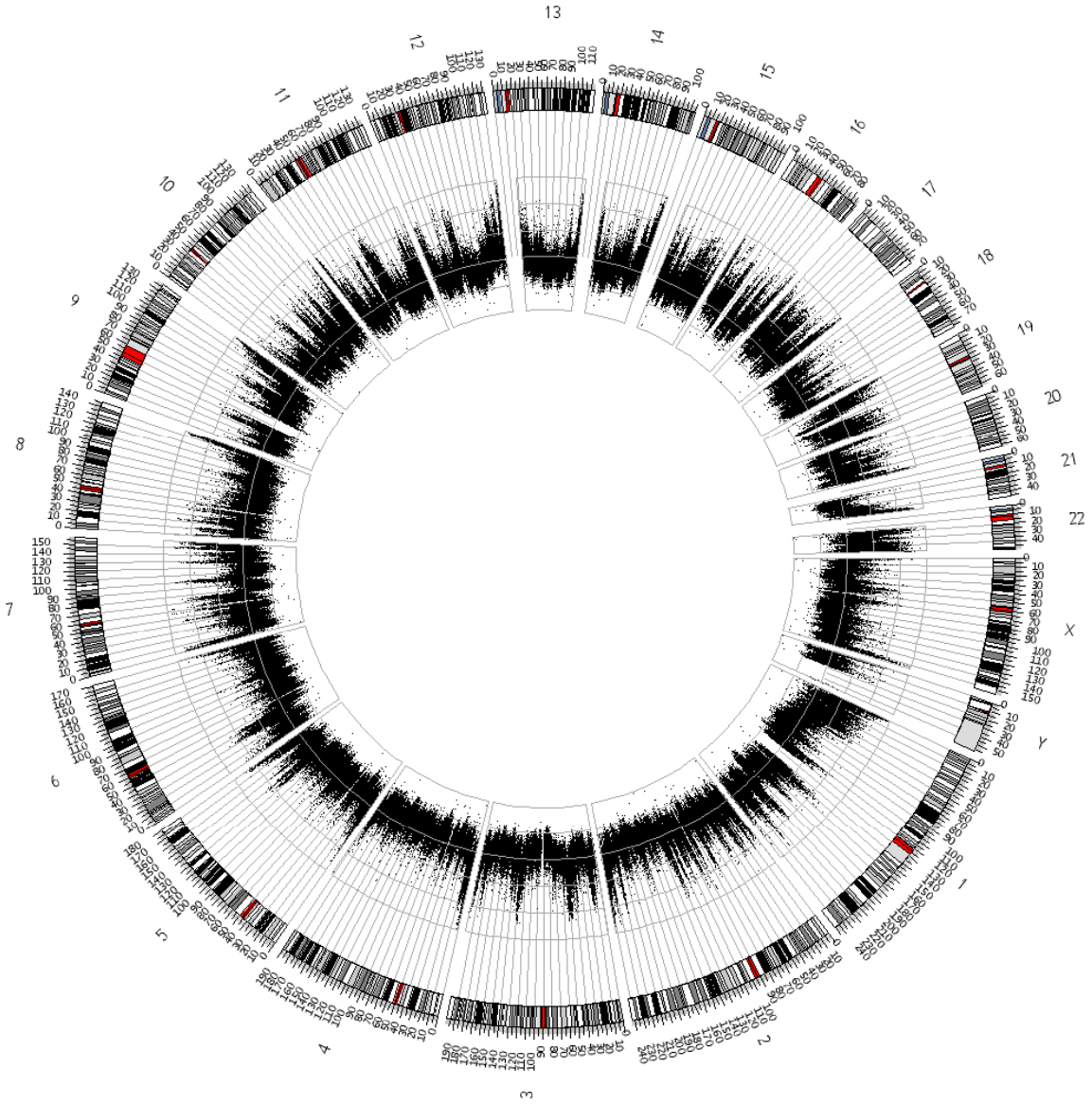
```
<rule link>
importance   = 80
rule         = min(_SIZE1_,_SIZE2_) < 2000
color        = dgrey
z = -5
</rule>
```



2D Plots

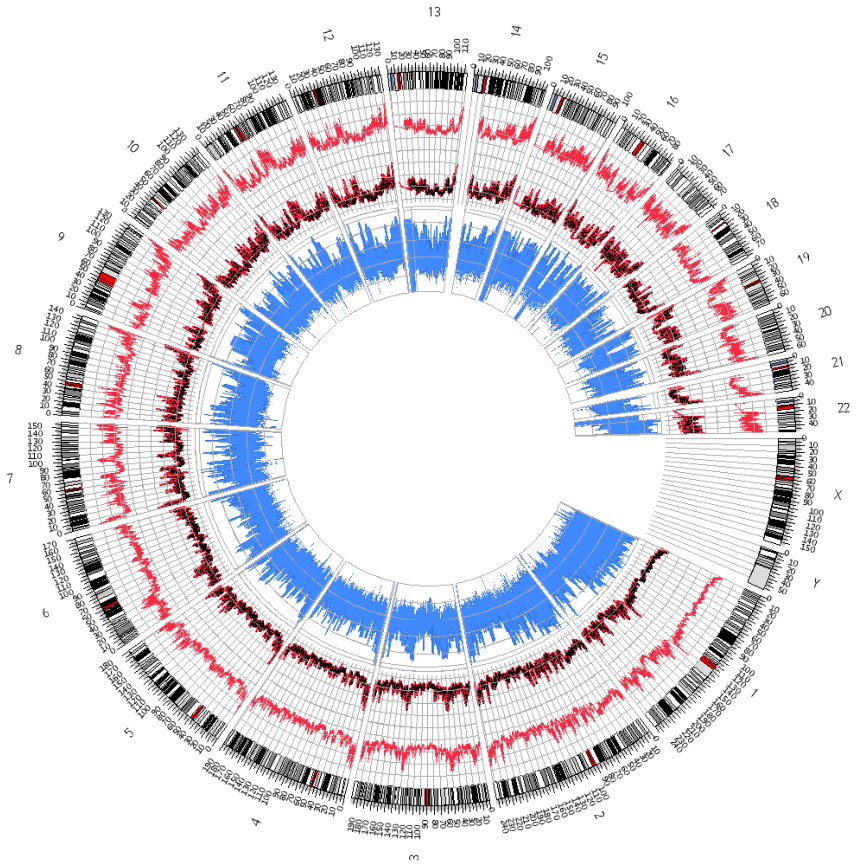
circos

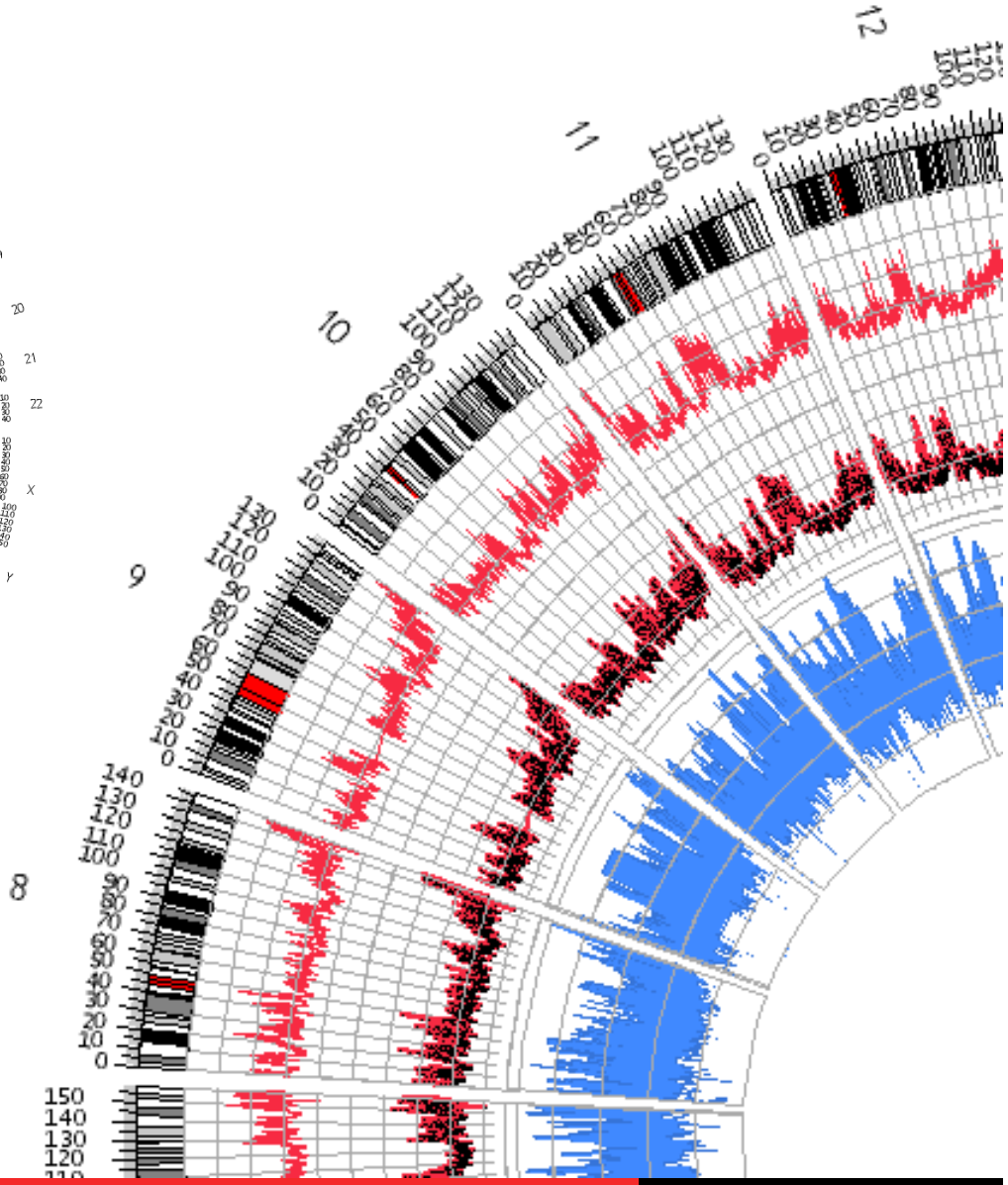
```
<plots>

<plot>

<data>
file  = gc.txt
size  = 1
color = black
type  = scatter
glyph = circle
</data>

orientation = out
offset   = -0.2
height   = 120
min      = 20
max      = 70
yspacing = 10
axes     = yes
axescolor = dgrey

</plot>

</plots>
```



2D Plots

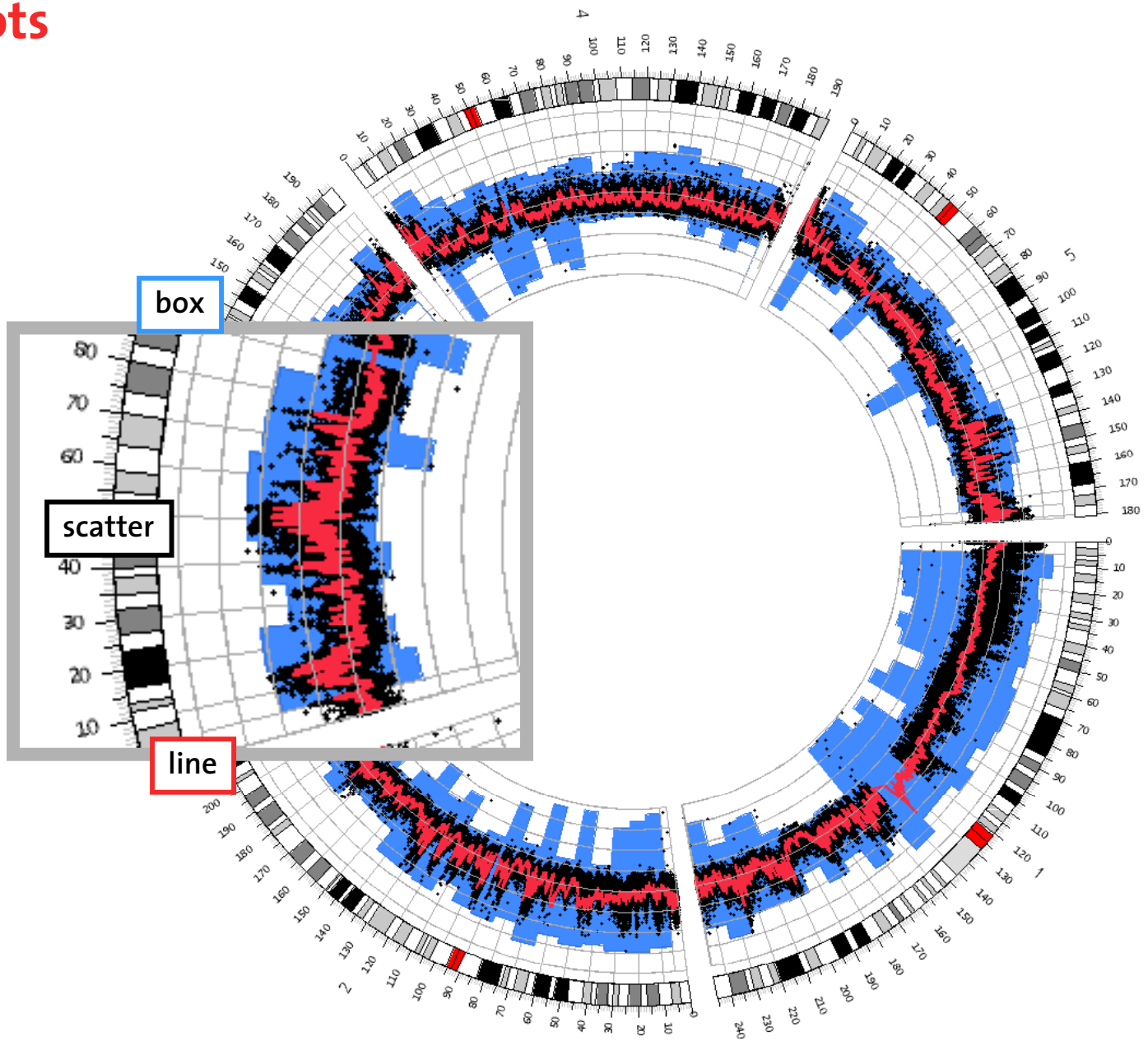
circos





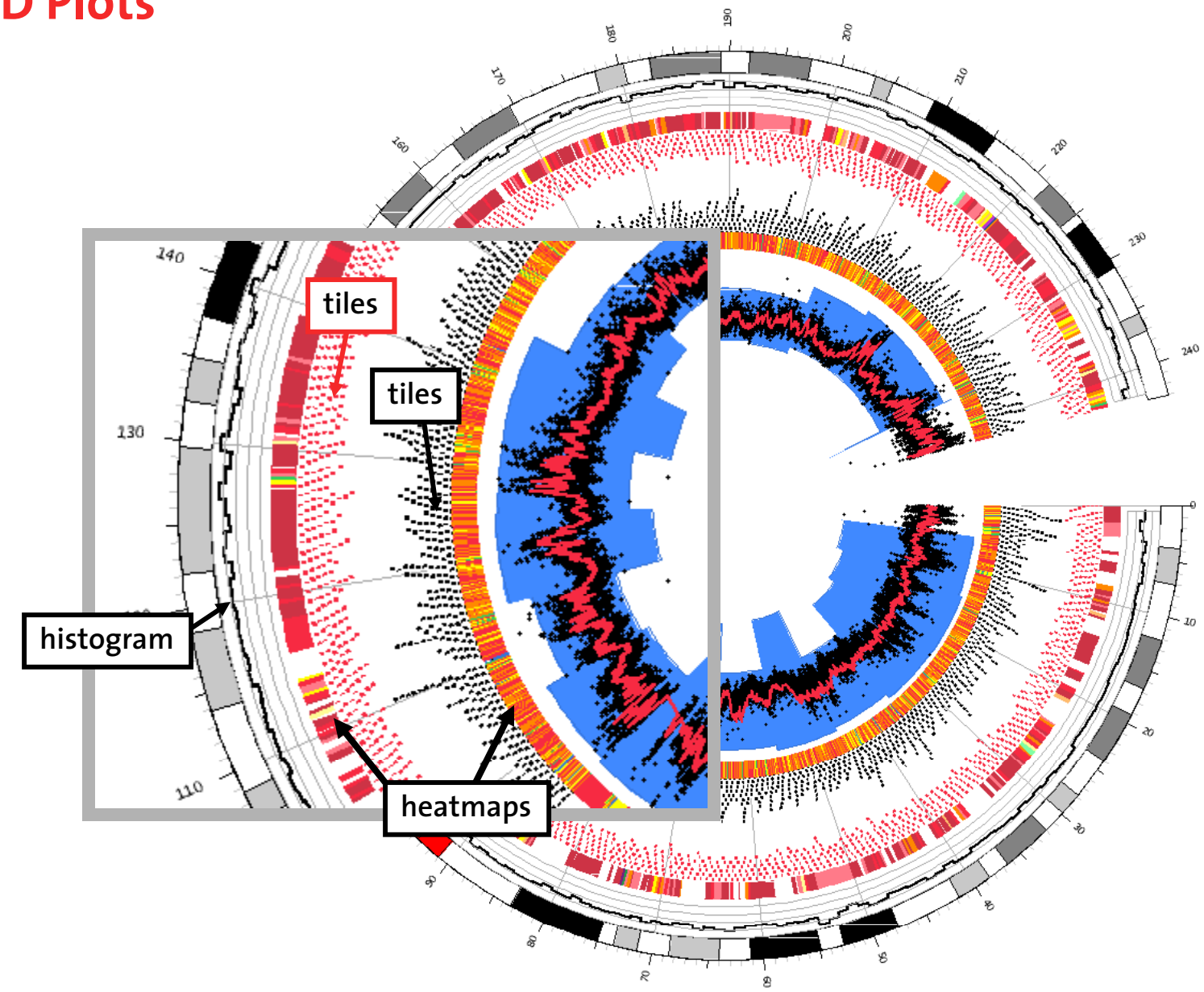
2D Plots

circos



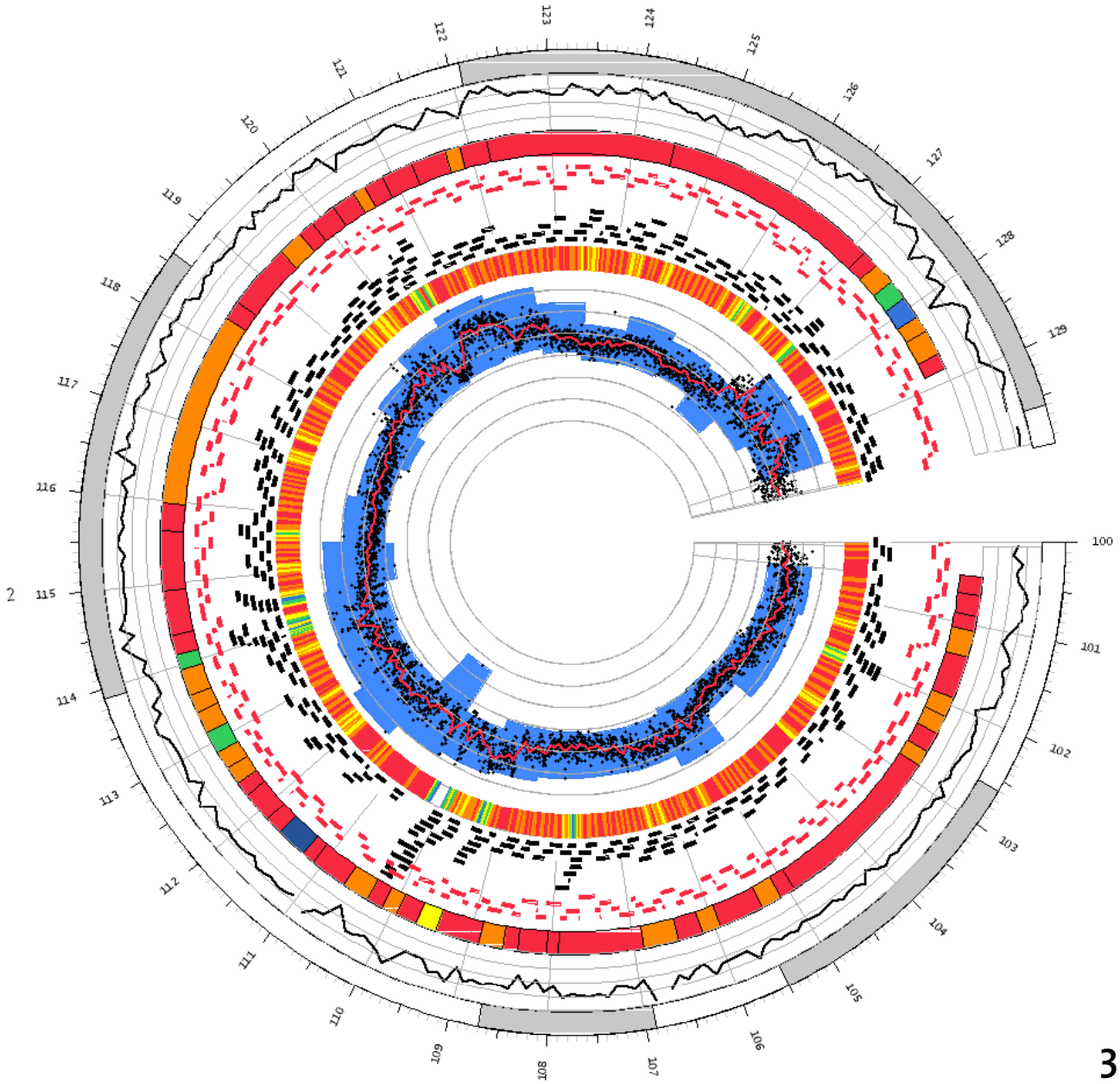
2D Plots

circos



2D Plots

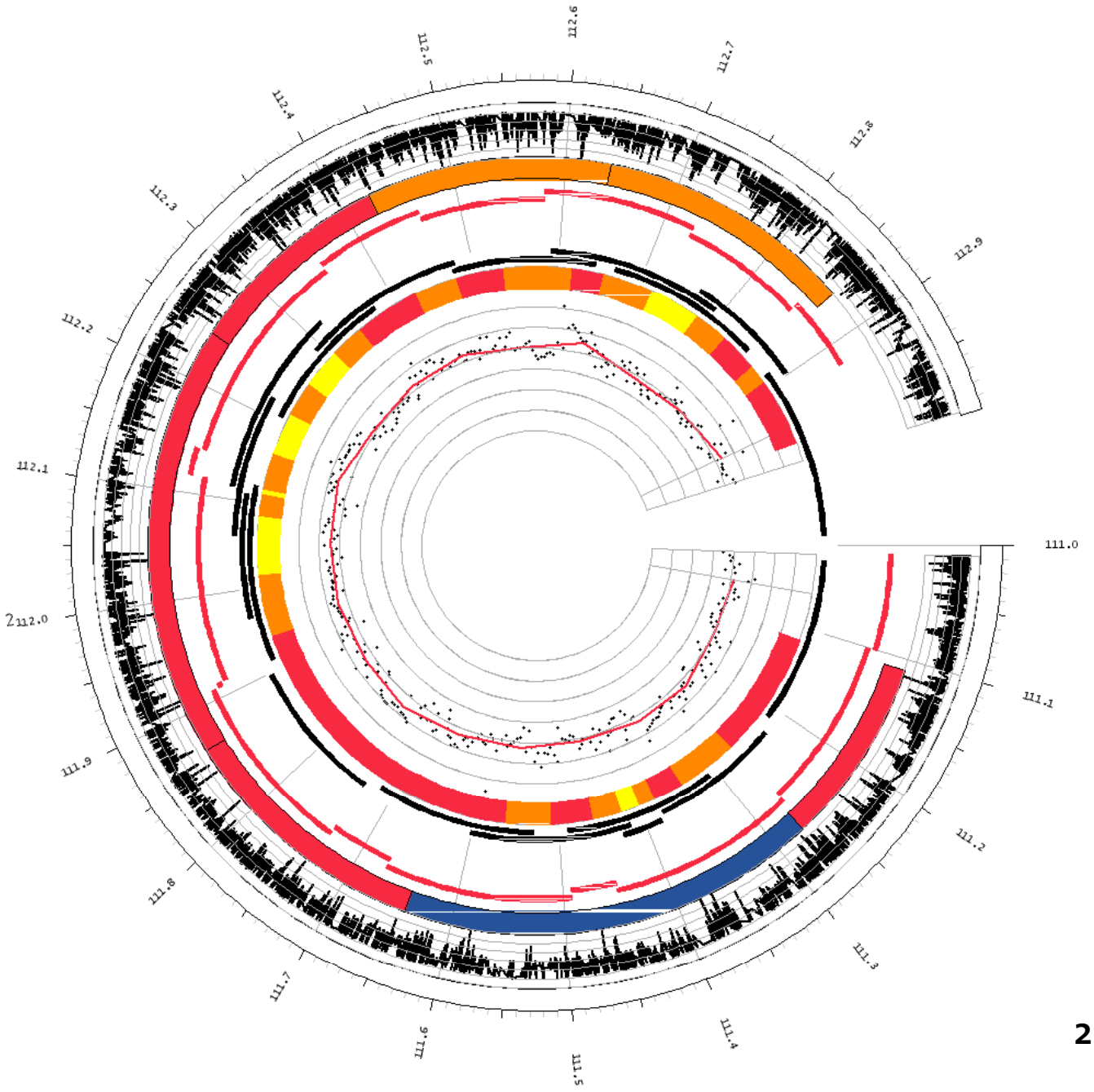
circos



30 Mb on chr2

2D Plots

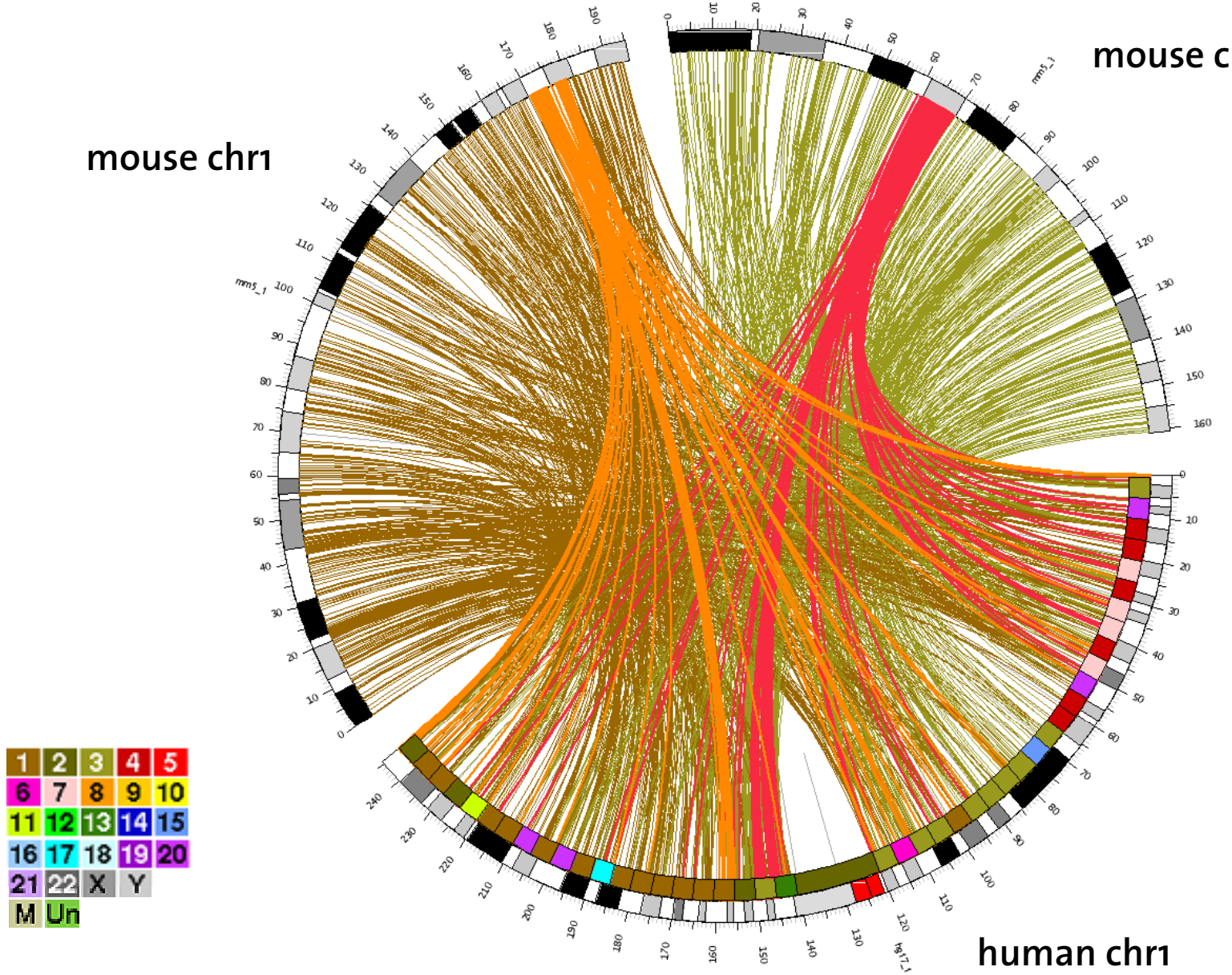
circos



2 Mb on chr2

Applications

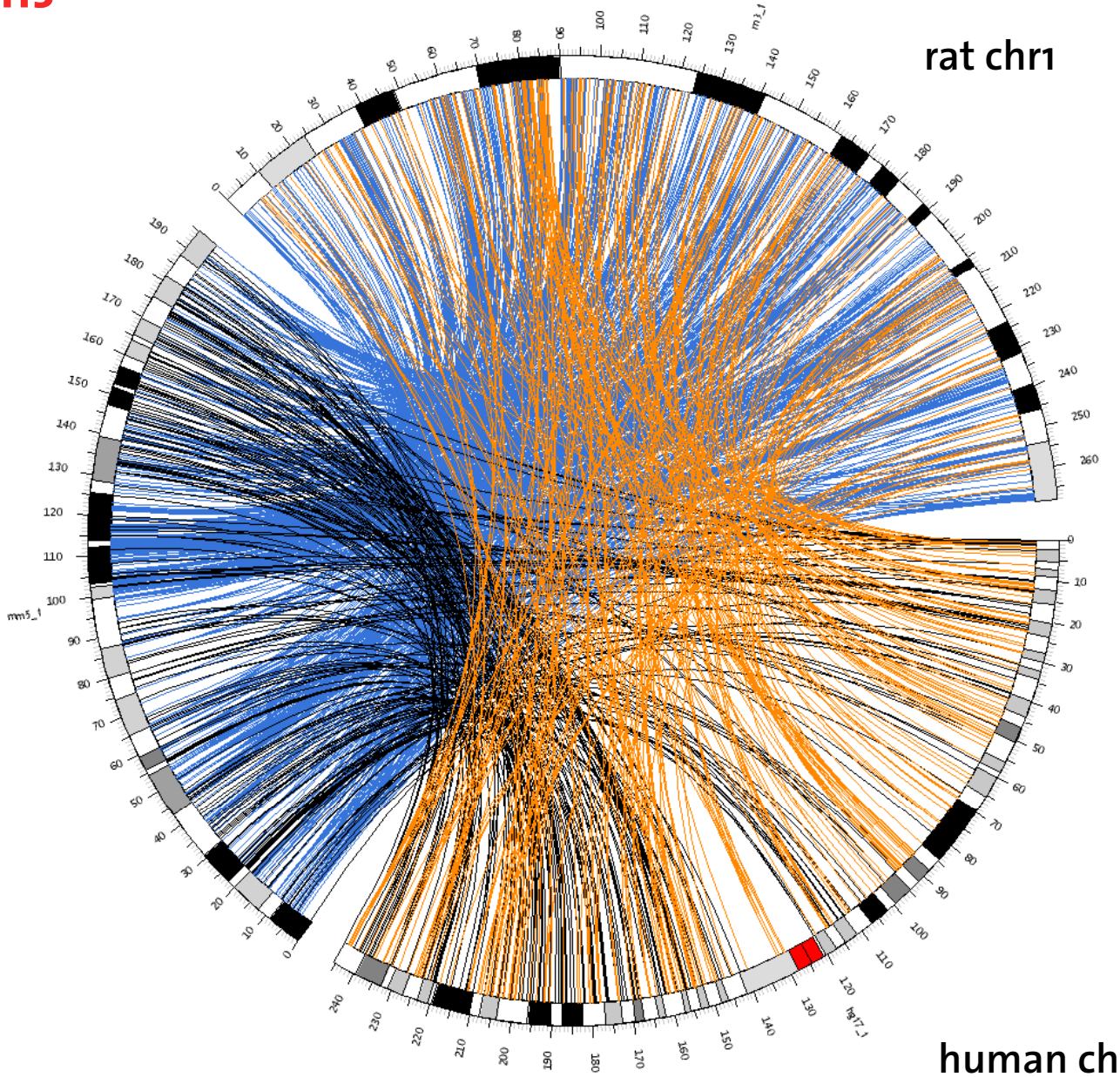
circos



Applications

circos

rat chr1

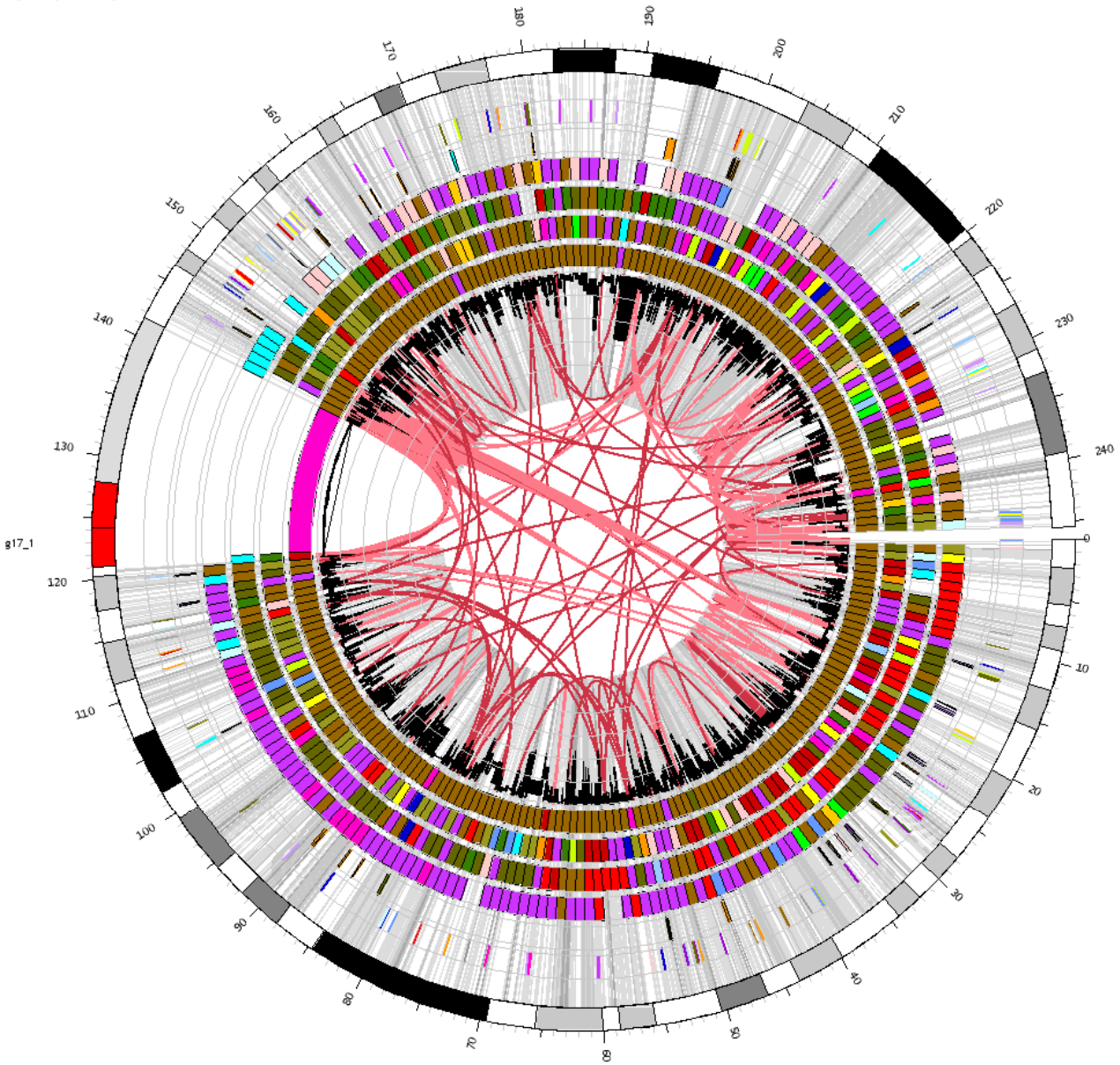


mouse chr1

human chr1

Applications

circos

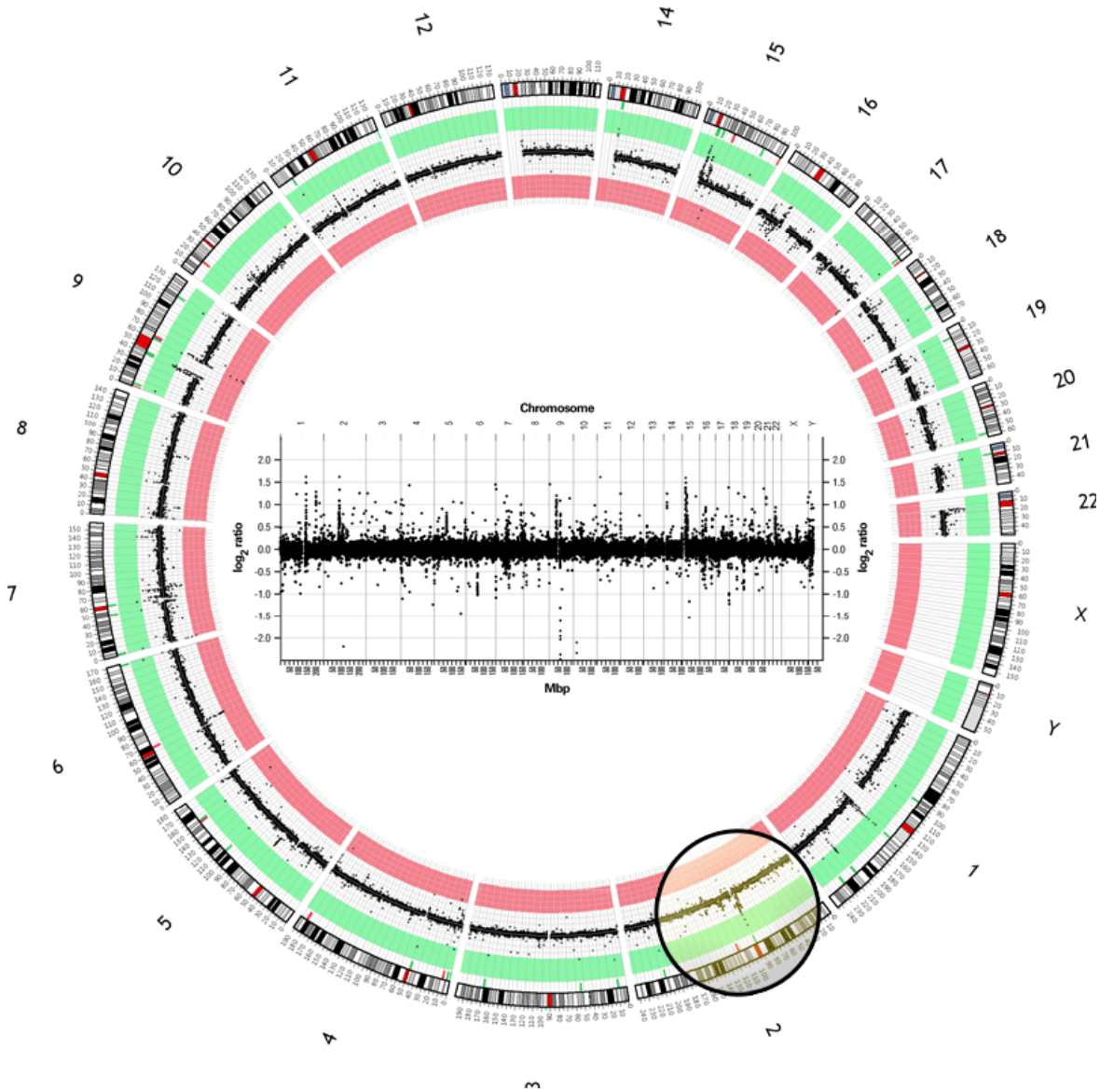


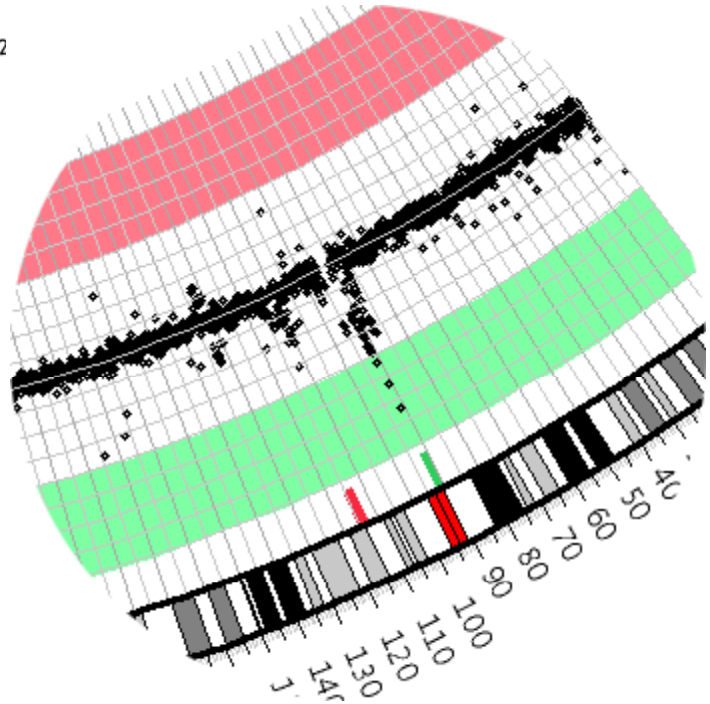
heat maps show conservation between human and

chimp (inner)
mouse
rat
dog
chicken
zebrafish (outer)

Applications

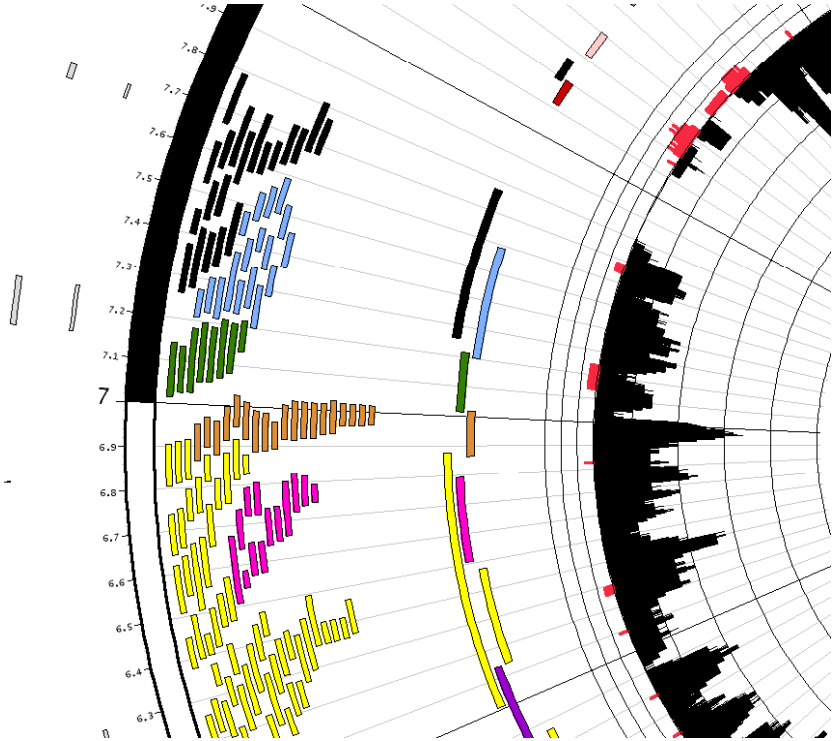
circos

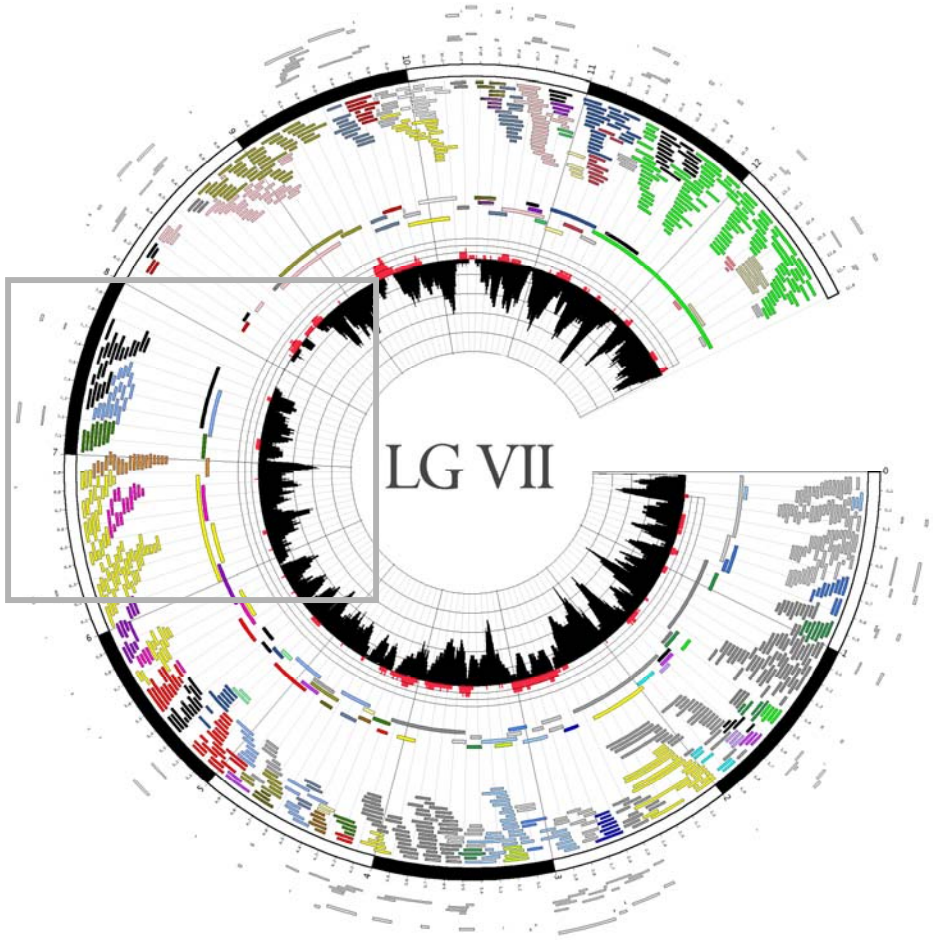




Applications

circos

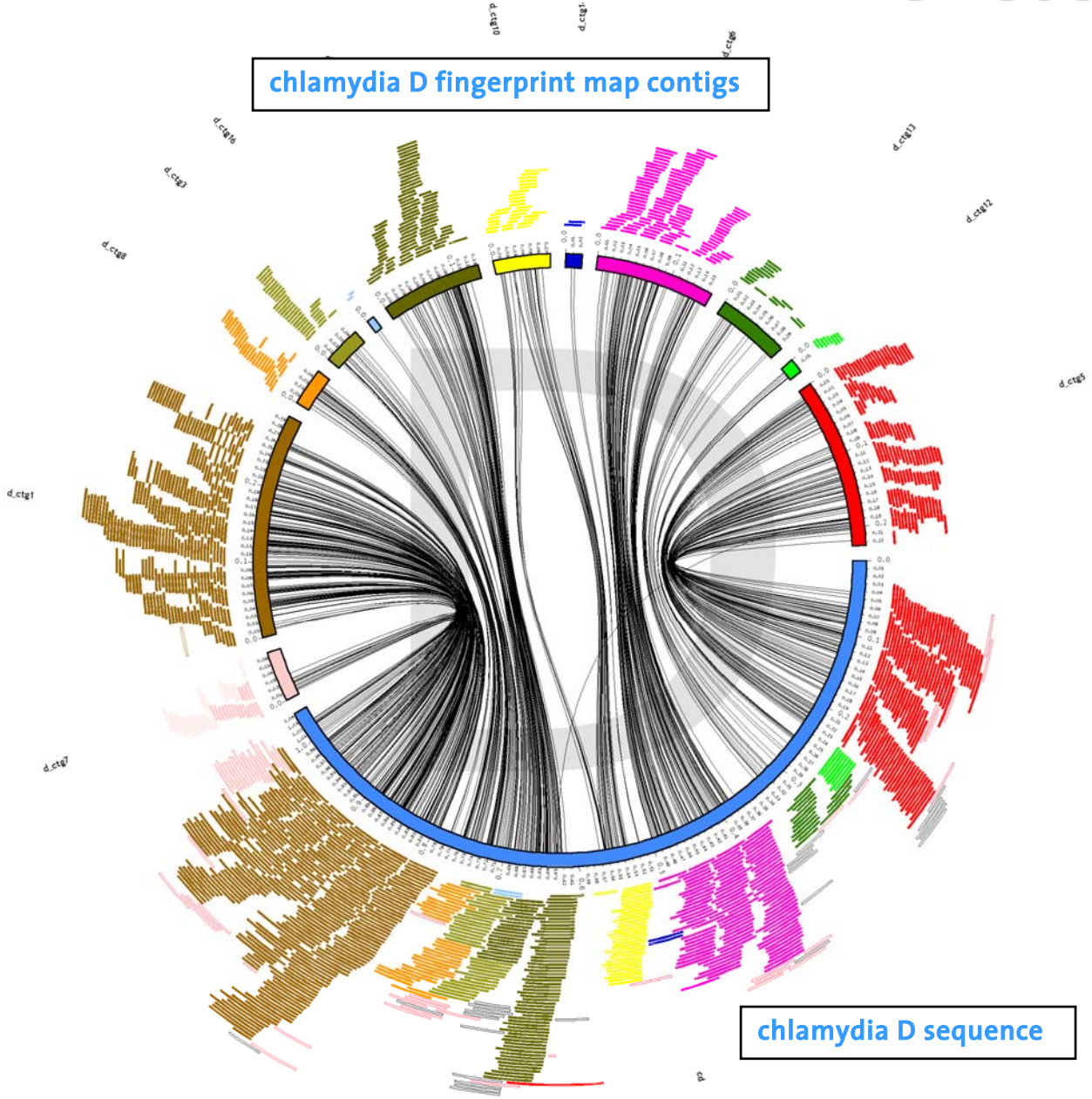




Applications

circos

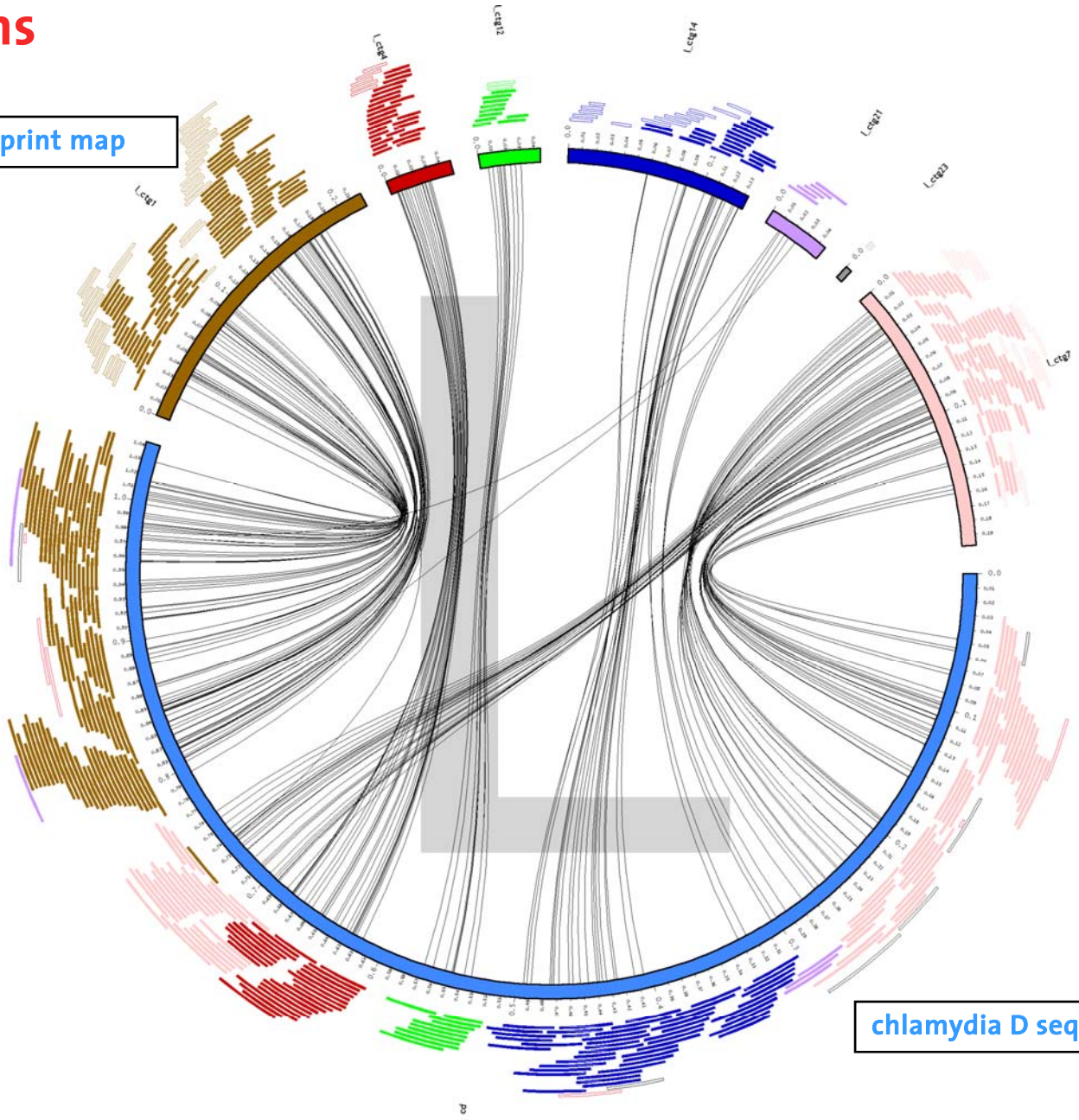
- fingerprint map clones localized on assembly by end sequence
- circle contains two independent entities: fingerprint map and assembly
  - lines join a clone's position in the map and in the sequence
- lack of cross-overs indicates consistency between map and sequence
  - map contigs ordered to minimize cross-over



Applications

circos

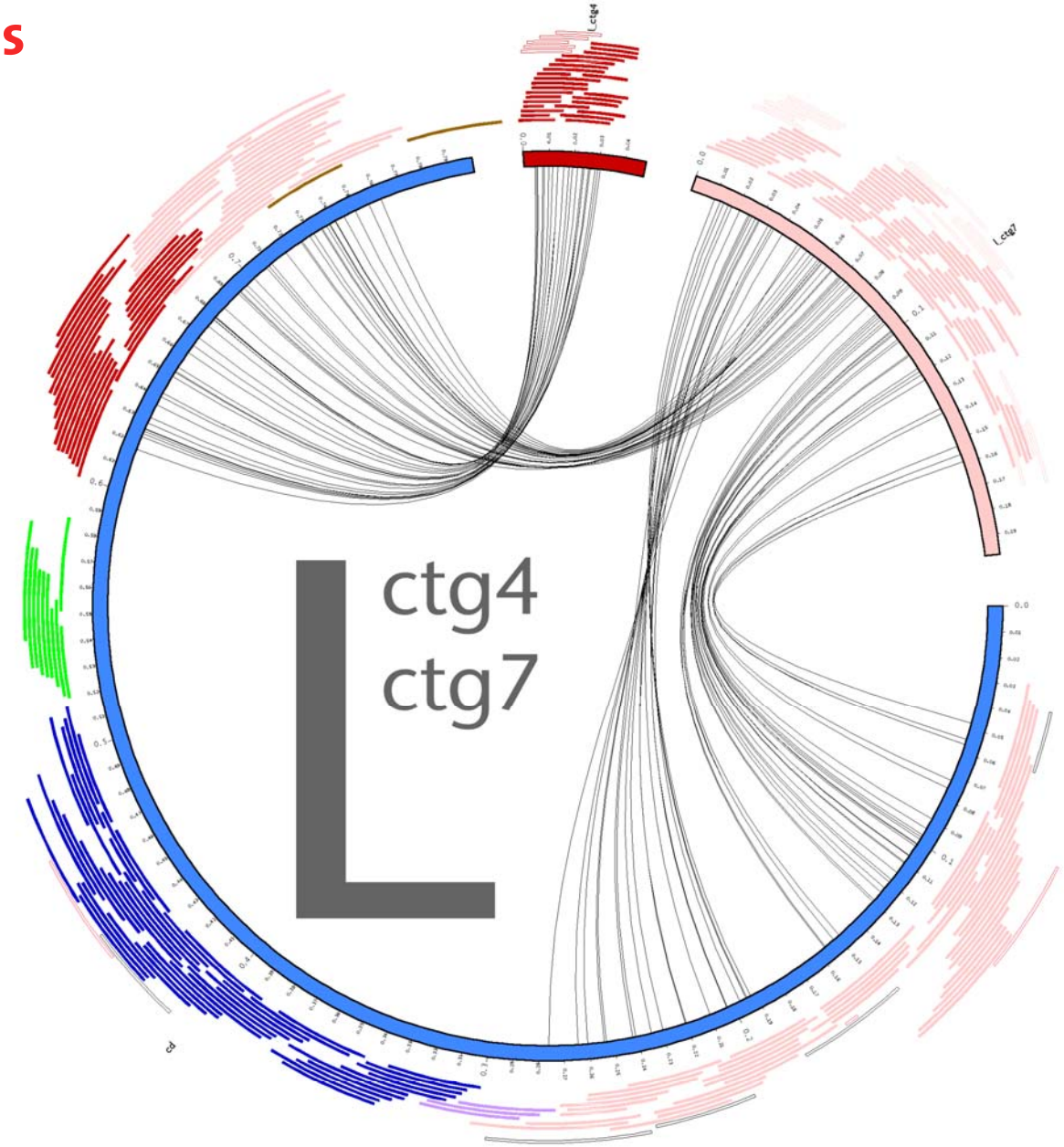
chlamydia L fingerprint map



chlamydia D sequence

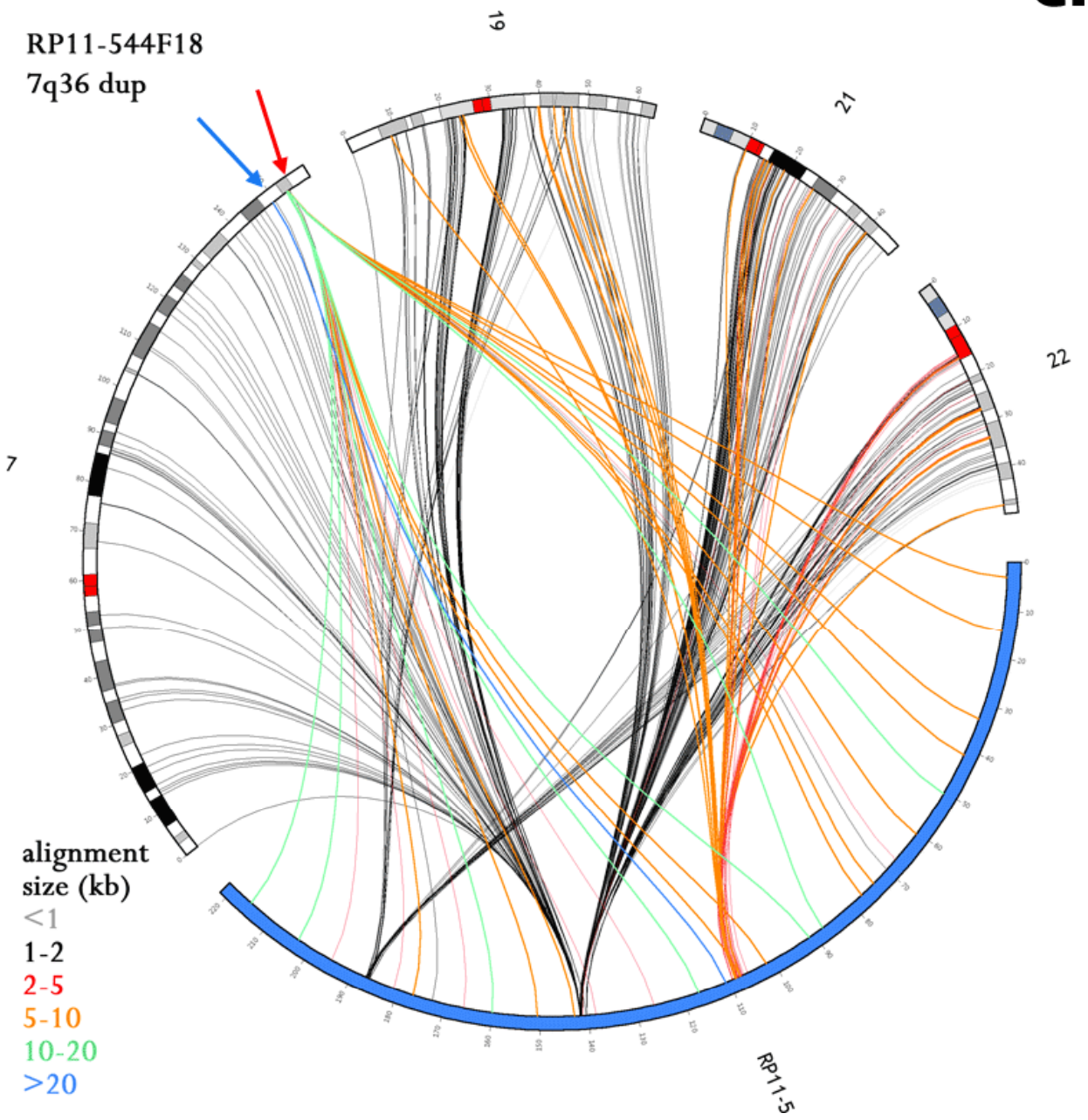
Applications

circos



Applications

circos

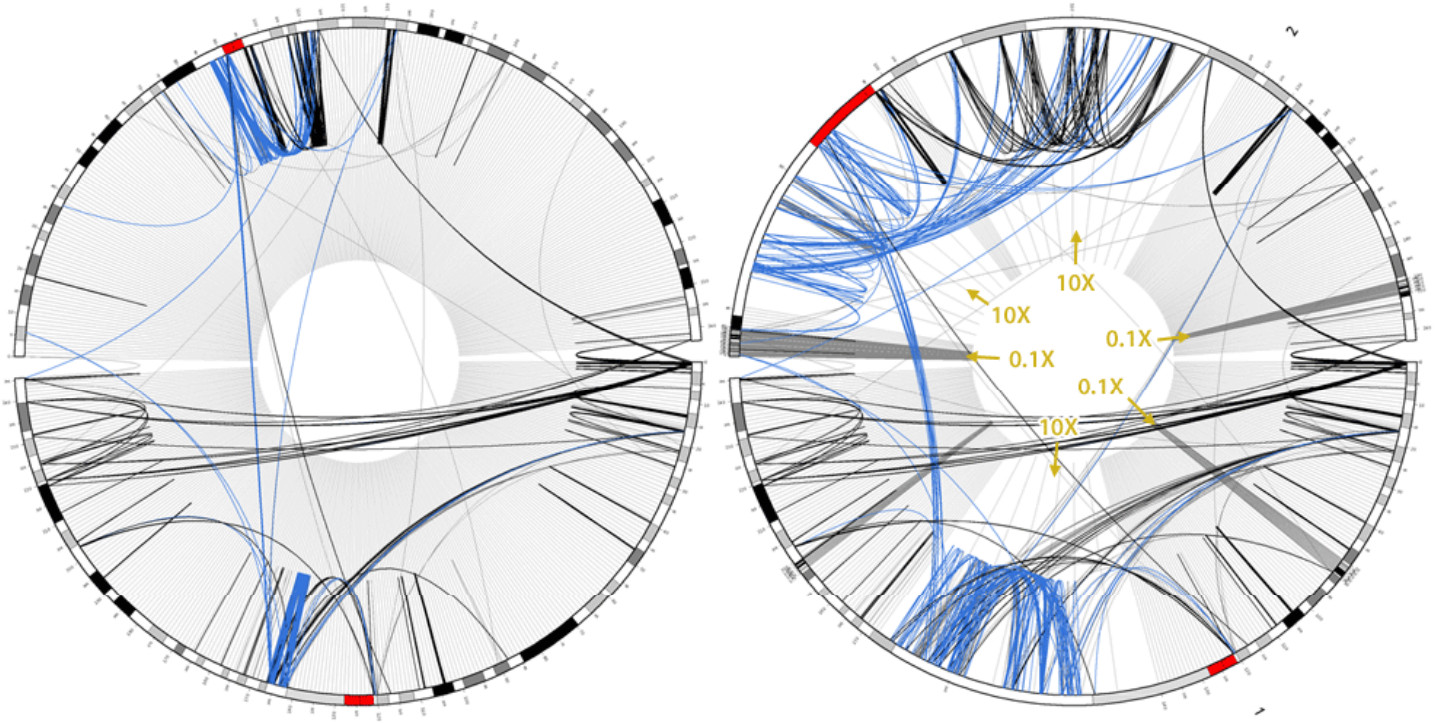


Non-Linear Scaling

circos

- genome is sparse
  - large deserts of no features
  - dense, distant groups of features
    - of course, depends on what features!

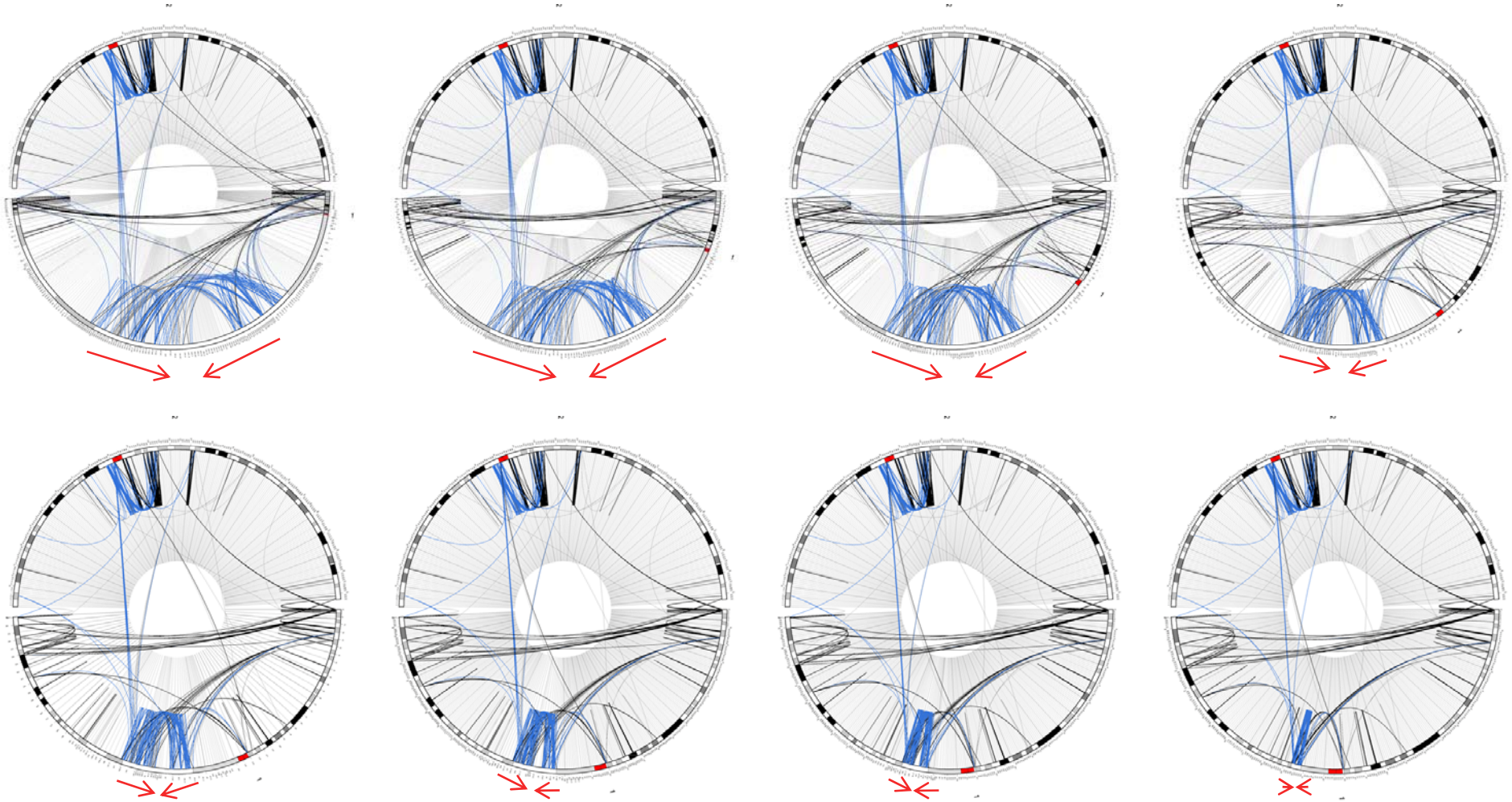
- Circos can locally expand/contract scale to zoom without cropping

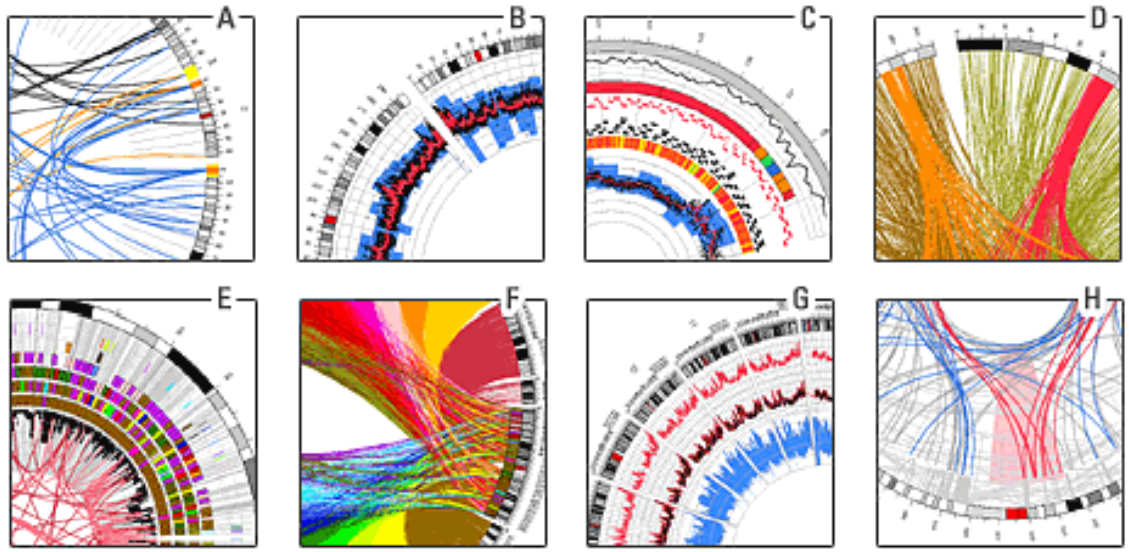


Non-Linear Scale

circos

local scale contraction →





Circos

round is good