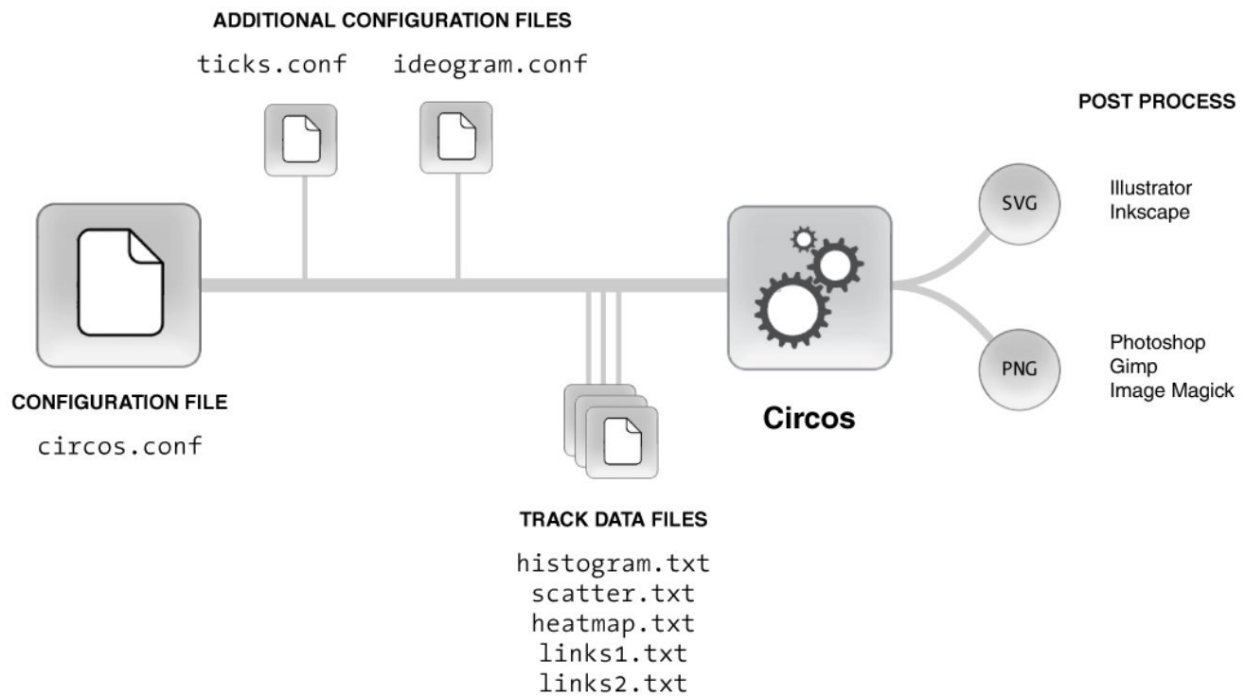


Configuration



File Organization

文件结构建议，对于不同的图片使用不同的文件目录。将配置文件放置到etc/中，数据放置在data/中，circos将自动查找主要配置文件circos.conf。

```
cancer snp/  
etc/  
  circos.conf  
  ticks.conf  
  ideogram.conf  
  ...  
data/  
  cnv.txt  
  genes.txt  
  ...
```

这样配置，就可以直接运行circos，无需添加 `-conf circos.conf` 参数。同时默认输出文件路径同circos.conf文件所在路径(`dir = conf(congifdir)`)，可修改：

```
circos -outputfile snp.png [-outputdir /tmp]
```

无需使用绝对路径，因为circos会在当前路径，配置文件路径，软件以及它们的相对几个路径查找主要配置文件和其中所需的数据文件，例如(不完整)：

```

dir1/dir2/dir3

where

dir1 = current_directory | configuration_directory | circos_binary_directory
dir2 = . | .. | ../.. | ../../../
dir3 = . | etc | data

```

The diagram illustrates the directory structure for Circos. It shows three variables: `dir1`, `dir2`, and `dir3`. `dir1` is defined as the concatenation of `current_directory`, `configuration_directory`, and `circos_binary_directory`. `dir2` is defined as a sequence of directory levels: `.`, `..`, `../..`, and `../../..`. `dir3` is defined as `.`, `etc`, and `data`. Red arrows point from the text 'where' to the definitions of `dir1`, `dir2`, and `dir3`. Another red arrow points from the word 'etc' in the `dir3` definition to the `etc` directory in the `dir2` definition.

circos.conf

配置文件格式，variable=value；简单定义在block中

```

<ideogram>
  thickness = 30p
  fill      = yes
  ...
</ideogram>

```

或多层blocks

```

<links>
<link>
  file      = data/set1.txt
  color     = black
  ...
</link>
<link>
  file      = data/set2.txt
  color     = red
  ...
</link>
</links>

```

The diagram shows a nested block structure for `<links>`. The `<links>` block contains two `<link>` blocks. The first `<link>` block has attributes `file = data/set1.txt` and `color = black`. The second `<link>` block has attributes `file = data/set2.txt` and `color = red`. Red arrows point from the `<link>` tag to the `file` attribute in the first block, and from the `</link>` tag to the `</link>` tag in the first block. Another red arrow points from the `<link>` tag to the `file` attribute in the second block. The `<links>` and `</links>` tags are highlighted with red boxes.

[配置文件参数][http://circos.ca/documentation/tutorials/configuration/configuration_files/]可指定4部分(<plots>, <plot>, data, <rule>), 重要程度依次递增, 后者可覆盖前者, 因此可在使用时相同类别可嵌套。

针对一些不需要更改的配置, 例如颜色和字体, 使用过程中保持默认配置就可, 使用 `<<include ...>>` 进行选择, `<<include CONFIG_FILE_PATH/CONFIG_FILE>>` :

Two files should always be imported from `etc/` in the Circos distribution. These are

```

# colors, fonts and fill patterns
<<include etc/colors_fonts_patterns.conf>>
# system and debug parameters
<<include etc/housekeeping.conf>>

```

The `etc/colors_fonts_patterns.conf` file itself imports several files from the Circos distribution.

若包含用户设置的路径, circos同样会在该路径下及对应相对路径搜索目标文件。

- + relative to the location of `CONFIG_FILE`
- + relative to `CONFIG_FILE_PATH/etc`
- + `CIRCOS_PATH/etc`
- + `CIRCOS_PATH/../etc`
- + `CIRCOS_PATH/..`
- + `CIRCOS_PATH`

同时可设置参数使用其他参数质

```
<block1>
<block2>
parameter1 = ...
</block2>
</block1>
```

you would use `conf(block1,block2,parameter1).`

使用eval进行perl脚本计算: `thickness = eval(1+1); eval(1.05 . "r")`

Color

`<colors>` 包含了所有颜色定义，最简单方式为导入默认颜色配置文件

`<<include etc/colors.conf>>` 包含默认设定好colors.brewer.conf, colors.uscs.conf, colors.hsv.conf

In turn, etc/colors.conf has the following definitions

```
# primary RGB colors
...

# Brewer palettes
# see etc/colors.brewer.conf
<<include colors.brewer.conf>>

# UCSC genome browser human chromosome colors
# see etc/colors.ucsc.conf
<<include colors.ucsc.conf>>

# HSV pure colors
# see etc/colors.hsv.conf
<<include colors.hsv.conf>>
```

Data Files

circos需要提供染色体定义文件，data tracks(<plot>blocks)文件，links(<link>blocks)文件和 highlights(<highlight>blocks)文件。

- KARYOTYPE, 名字，标签，起点，终点，颜色

```
chr - axis1 1 0 1000 black
chr - axis2 1 0 1500 blue
chr - axis3 1 0 2000 green
```

- LINE, SCATTER, HISTOGRAM, HEATMAP图对应KAYOTYPE的位置信息

```
#chr start end value [options]
hs5 50 75 0.75
```

- TEXT, 定义染色位置相关的内容信息

```
#chr start end label [options]
hs5 50 75 ABC
```

- CONNECTOR, 连接相同染色体上的两个位置, 起点终点必须在同一染色体

```
#chr start end [options]
hs5 50 1500
```

- LINKS, 连接相同或不同染色体的两个相关区间

```
# chr1 start1 end1 chr2 start2 end2 [options]
hs1 200 300 hs10 1100 1300
hs7 50 150 hs 5000 6000 color=blue
```

Runtime Parameters

命令行使用-debug_group选项选择输出运行内容, 显示所有输出内容名称

```
circos -debug_group
```

增加输出其他内容

```
circos -debug_group +timer,+io
```

或指定输出/全部输出

```
circos -debug_group timer
```

```
circos -debug_group _all
```

命令行调整输出文件路径和名称-outputdir, -outputfile, 覆盖配置文件输出; 这些参数一般在<image>模块中

```
<image>
dir = /path/to/your/output/directory
file = yourimage.png
...
</image>
```

查看所有内部配置树状信息

```
circos -cdump
```

```
circos -cdum | grep default_color
```

```
default_color => "black"
```

```
circos -cdump ideogram
```

命令行修改配置文件参数值, 以下将修改模块<block1><block2 >中的name值

```
circos -param block1/block2/.../name=value
```

```
circos -param karyotype=myfile.txt ...
circos -param image/radius=500p
circos -param ideogram/show=no
```

PNG/SVG输出

默认条件下输出为：3000x3000pixels，白色背景， circos.png文件。默认包含在：

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

```
# etc/image.conf
<<include image.generic.conf>>
<<include background.white.conf>>
```

```
# image.generic.conf
dir                = .
file               = circos.png
png                = yes
svg                = yes
radius             = 1500p
angle_offset       = -90
#angle_orientation = counterclockwise
auto_alpha_colors  = yes
auto_alpha_steps   = 5
```

```
# background.white.conf
background         = white
```

angle_offset=-90，表示第一个idogram出现位置，这里为图像上部

若需要修改覆盖，使用*后缀语法

```
# circos.conf
<image>
<<include etc/image.conf>>
file*   = myfile.png
radius* = 1000p
</image>
```

SVG为scalable vector graphics, 使用基于向量的元素绘图，例如线条，圆形，方形等，而不是使用像素作图(PNG)。默认条件下输出，可使用命令行参数抑制

```
circos -nosvg ...
```

QUICK START

Karyotype

karyotype文件是必须的，该文件定义了染色体的名称，大小和颜色，同时又不仅限于染色体信息，还可以提供序列contigs, genes, indexed positions, blocks of time, 任何整数构成的坐标系统。circos软件内含多个常见序列装配信息：人类，大鼠，小鼠，果蝇，这些文件位于data/karyotype中。

简单绘制染色体图，需提供karyotype文件，图像的最大像素为1500，如果radius=0.8，就意味着 $0.8 \times 1500 = 1200$ 像素，染色体之间距离通过spacing来设置：

```
# circos.conf

karyotype = data/karyotype/karyotype.human.txt

<ideogram>

<spacing>
default = 0.005r
</spacing>

radius      = 0.9r
thickness   = 20p
fill        = yes

</ideogram>
```

还需以下三个为绘图标准且必须配置内容

```
<image>
# Included from Circos distribution.
<<include etc/image.conf>>
</image>

# RGB/HSV color definitions, color lists, location of fonts, fill patterns.
# Included from Circos distribution.
<<include etc/colors_fonts_patterns.conf>>

# Debugging, I/O and other system parameters
# Included from Circos distribution.
<<include etc/housekeeping.conf>>
```

以上绘制简单框架图像，不含任何文字信息

Ticks & Labels

向ideogram模块添加标签参数，刻度标签信息定义在ticks内。另外，在定义ideogram位置时，参考信息为image半径，当定义track位置时，参考位置为ideogram半径。

```
karyotype=data/karyotype/karyotype.human.txt #定义ideogram文件
```

```
chromosomes_units=10000000 #定义染色体单位大小，以后使用单位为u
```

chromosomes_units=10,000,000, ticks的spacing在10M, 5M, 1M分别为10u, 5u和1u

label信息位于<ideogram>内，label_parallel表示label方向，yes，和染色体方向平行；no，和染色体方向垂直：

```
# Minimum definition for ideogram labels.

show_label      = yes
# see etc/fonts.conf for list of font names
label_font      = default
label_radius     = 1r + 75p
label_size      = 30
label_parallel   = yes
```

ticks信息位于<ticks>内

```

show_ticks      = yes
show_tick_labels = yes

<ticks>

radius          = 1r
color           = black
thickness       = 2p

```

Ideogram Selection, Scale, Color & Orientation

默认情况下显示所有染色体，通过参数chromosomes_display_default=no来选择部分染色体显示

```
chromosomes=hs1;hs2;hs3;hs4
```

```
chromosomes=/hs[1-4]$/ #正则表达
```

```
chromosomes=/hs[1-4]$/;hs10;hs11
```

```
chromosomes=/hs[1-4]$/;-hs3 #排除hs3
```

```
chromosomes=hs1:(-100,120-);hs2;hs3;hs4 #匹配染色体对应区间
```

使用绝对或相对放大来修改ideogram尺寸

Absolute scale applies a fixed magnification to an ideogram

```

# hs1 0.25x zoom
# hs2 2.00x zoom
chromosomes_scale = hs1=0.25,hs2=2.0

```

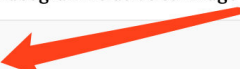


Relative scale defines the size of the ideogram relative to image circumference

```

# hs1 25% of figure
# hs2 50% of figure
chromosomes_scale = hs1=0.25r,hs2=0.50r

```



自定义染色体颜色

```
chromosomes_color=hs1=red;hs2=orange;hs3=green;hs4=blue
```

或使用*后缀覆盖默认定义值(colors_fonts_patterns.conf)

```

<colors>
chr1* = red
chr2* = orange
chr3* = green
chr4* = blue
</colors>

```

Links & Rules

links通过曲线或直线代表了两染色体位置之前的相关性(相似性，差异性，融合等)，使用link模块添加修改。


```

<links>
<link>
file          = data/5/segdup.txt
radius        = 0.8r
bezier_radius = 0r
color         = black_a4
thickness     = 2
</link>
</links>

```



segdup.txt文件格式

```

1 hs1 465 30596 hs2 114046768 114076456
2 hs1 486 76975 hs15 100263879 100338121
3 hs1 486 30596 hs9 844 30515
4 hs1 486 9707 hsY 57762276 57771573
5 hs1 486 9707 hsX 154903076 154912373
6 hs1 486 9707 hs16 427 9533

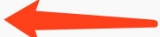
```

可在任何模块内添加rule模块，用于改变当前数据(links, histogram bins, scatter plot glyphs等)。每个rule都含有一个状态，格式声明和可选的flow声明。依次评估rules内的rule，如果当前rule正确，则执行，不再检测接下来的rules(除非 flow=continue)，假如当前rule条件错误，则接着查看下一个rule。

```

<rule>
condition    = var(intrachr)
# Any links that are intra-chromosomal will not be shown. Further rules are not tested.
show        = no
</rule>

```



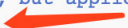
如果起点或终点在hs1，则改变link的起点和终点的半径

```

<rule>
# If the link's start is on hs1...
condition    = from(hs1)
# ...set the radial position of the link's start to be close to the ideogram.
radius1      = 0.99r
</rule>

<rule>
# Same as the rule above, but applies to the end of the link.
condition    = to(hs1)
# 'radius2' (like chr2, start2, end2) refers to the variable 'radius' of the end of the link.
radius2      = 0.99r
</rule>

```



默认条件下当前rule为假才继续执行

```
flow=continue if false
```

修改，强制继续执行或强制停止

```
flow=continue{if true|false}
```

```
flow=stop{if true|false}
```

Histograms

除了link图外，可使用line plots, scatter plots, histograms和heatmaps来描述位置关系

line, scatter, heatmap, histogram图数据格式一样


```
#chr start end value [options]
...
hs3 196000000 197999999 71.0000
hs3 198000000 199999999 57.0000
hs4 0 1999999 28.0000
hs4 2000000 3999999 40.0000
hs4 4000000 5999999 59.0000
...
```


可选参数可为该数据点的一个相关参数，可以时颜色或名称

```
hs3 196000000 197999999 71.0000 fill_color=blue
```

使用参数，例如id可以连接rule用于选择修饰数据点

```
# in data file
hs3 196000000 197999999 71.0000 id=abc

# in rule block
<rule>
condition = var(id) eq "abc"
fill_color = blue
</rule>
```



每一个histogram都应定义在<plot>模块中，使用histogram时，需要定义type和file参数，还有内半径r0，外半径r1，相对值为r(radius)绝对值为p(pixels)。

```
type = histogram
file = data/5/segdup.hs1234.hist.txt
r1   = 0.88r
r0   = 0.81r
```

同时histogram拥有fill和ouine参数，默认参数位于etc/tracks/*.conf，同时可以选择orientation=in|out

```
# etc/tracks/histogram.conf
color          = black
thickness      = 1
r1             = 0.89r
r0             = 0.8r
orientation    = out
```

Axes & Backgrounds

背景可使用backgrouds模块直接和axes模块渐变配色，同时都得在plot模块内

Axes类似ticks，按照组来定义，每组可指定相对或绝对的空间，axes对应参数

- + **spacing** - absolute or relative spacing of the axis
- + **position** - fixed position (or positions) for axes
- + **position_skip** - fixed position (or positions) to skip when drawing axis lines
- + **y0** - absolute or relative start of axis lines
- + **y1** - absolute or relative start of axis lines
- + **color** - color of axis lines
- + **thickness** - thickness of axis lines

axes就是在半径位置上绘制颜色条带

```
<axes>

# Show axes only on ideograms that have data for this track
show = data

thickness = 1
color     = lgrey

<axis>
spacing   = 0.1r
</axis>

<axis>
spacing   = 0.2r
color     = grey
</axis>

<axis>
position  = 0.5r
color     = red
</axis>

<axis>
position  = 0.85r
color     = green
thickness = 2
</axis>

</axes>
```

background单元着色范围在y0至y1半径范围内

```
<backgrounds>

# Show the backgrounds only for ideograms that have data
show = data

<background>
color = vvlgrey
</background>
<background>
color = vlgrey
# the "r" suffix indicates position relative to track data range
y0    = 0.2r
y1    = 0.5r
</background>
<background>
color = lgrey
y0    = 0.5r
y1    = 0.8r
</background>
<background>
color = grey
# if y1 is not specified, the plot maximum is used (e.g. y1=1r)
y0    = 0.8r
</background>

</backgrounds>
```

highlight数据格式：

```
1 hs1 0 247249719
2 hs2 0 242951149
3 hs3 0 199501827
4 hs4 0 191273063
5 hs5 0 180857866
6 hs6 0 170899992
```

Heat Maps

Heatmap用于展示基因的一个位置或区域和一个值的关系。

该图线性线性将一个范围值[min, max]映射到一组颜色中c[n], i=0..N

```
f = (value - min) / ( max - min )
n = 0      if f < 0
    N      if f > 1
    N * f   otherwise
```

heatmap参数和histogram参数格式形同

```
<plot>
type  = heatmap
file  = data/5/segdup.hs1234.heatmap.txt
r1    = 0.89r
r0    = 0.88r
</plot>
```

如果使用scale_log_base, 那么比对就不是线性了

```
n = N * f**(1/scale_log_base)
```

假如scale_log_base大于1, 那么颜色动态范围将在接近最小值初延伸, 否则在接近最大值处延伸。

heatmap的颜色默认定义在etc/colors.conf文件中。本次使用的比对颜色范围由5个水平的染色体颜色透明程度构成, 外加一个不透明颜色。

```
color = hs1_a5,hs1_a4,hs1_a3,hs1_a2,hs1_a1,hs1
```

Text

文本信息描述了一个基因组位置和一个文本之间的关系。和其他tracks一样

```
<plot>
type  = text
file  = data/6/genes.labels.txt
r1    = 0.8r
r0    = 0.6r
...
</plot>
```

但需指定字体和标签大小, 同时可使用padding和rpadding来设置文本边界

```
label_font = light
label_size = 12p
rpadding   = 5p
```

2D DATA TRACKS

所有的2D图形都定义在plot模块中，一个plot模块内，一个图。每个plot模块中，定义图形格式，文件，还可以插入包含多重rule的rules模块。

1. [Scatter Plots](#)
2. [Line Plots](#)
3. [Histograms](#)
4. [Tiles](#)
5. [Heat Maps](#)
6. [Text—Basic](#)
7. [Text—Stacking](#)
8. [Text—Rules](#)
9. [Glyphs—Part I](#)
10. [Glyphs—Part II](#)
11. [Connectors](#)
12. [Putting It All Together](#)

SCATTER PLOTS

每个类型图都有其默认参数设定，位于etc/tracks内。2D图均有相同的格式参数：

```
<plots>

<plot>

  show = yes
  type = scatter

  file = data/6/snp.density.txt
  r1    = 0.75r
  r0    = 0.90r
  max   = 1.0
  min   = 0.0

  glyph      = rectangle
  glyph_size = 8
  color      = red
  stroke_color = dred
  stroke_thickness = 1

</plot>

</plots>
```

show: 和highlights或links一样，决定是否绘制该图

type: 图形类型，scatter，line，histogram或heatmap等

file: 含图形数据的文件

min/max: 图形轴的范围值，超过该范围将被切除

r0/r1: 图形轨迹的内外半径，和lightlights一样可以是绝对值或相对值

glyph: scatter图使用的图形状参数，可以是circle，rectangle或triangle

glyph_size: glyph大小，单位为pixels，像素

color: 如果使用，glyph将会为该颜色

stroke_color: 如果使用，glyph轮廓将会为该颜色

stroke_thickness: 如果使用, glyph的轮廓将使用该thickness, 单位pixels

2D文件格式:

The data file format for 2D plots is

```
chr start end value options
```

Note that a value is associated with an explicit span, and not a single base pair position. For example,

```
# value 0.005 at span 1000-2000
hs1 1000 2000 0.005
# value 0.010 at span 2001-2001, e.g. a single base position
hs1 2001 2001 0.010
```

plot背景, 使用background模块添加, 可同时添加多个背景模块, 使用条纹和分层填充背景, y0/y1分别对应了填充的半径范围。

```
<backgrounds>
<background>
color      = vlgrey
# absolute fill range
y1         = 0.5
y0         = 0
</background>
<background>
color      = vlred
# relative fill range
y1         = 1r
y0         = 0.75r
</background>
</backgrounds>
```

plot轴, 可在plot中使用axes绘制y轴网格线, 对应给定spacing的同心圆弧

```
<axes>
<axis>
color      = grey
thickness  = 1
spacing    = 0.05r
</axis>
</axes>
```

LINE PLOTS

线图不同于散点图在于, 每个点没有图形, 而是通过线将其连接。线图格式和散点图类似, 有以下区别:

type: type=line

stroke_thickness: 对应为thickness

stroke_color: 对应为color


max_gap: 临近点距离大于max_gap, 经不会使用线连接, 避免绘制线图跨越大的gaps

同时, 落在min/max范围以外的数据点将会绘制在min/max的边缘; 若想去除某些带数据点, 使用rule模块, 设置show=no, 例如超过min/max 0/0.5的点:

```

<rules>
<rule>
condition = var(value) < 0 || var(value) > 0.5
show      = no
</rule>
</rules>

```



Histograms

展示线图上的点，临近点直线连接。创建histogram，设置type=histogram。在线图和散点图中，数据点位于区间跨度的中点：

hs1 1000 2000 0.5，绘制点位于1500的位置；而对于histogram，整个范围1000-2000(bin)的值都为0.5。

若设置extend_bin=yes，那么该bin的左右将会延伸至相邻bin的中间位置，该设置为默认设置(extend_bin=yes)，设置extend_bin=no将不会延伸bins。


假如数据密度很大，那么histogram将会变得很密集难以区分，建议使用线图。

使用rule模块来跳过明确的数据点。例如，在每1Mb区间绘制histogram

```

<rule>
condition = var(start) % 1Mb
show      = no
</rule>

```




使用skip_run=yes来避免读取所有数据，设置改参数，circos将仅读取相同数值的连续数据点的第一个点：

```

<plot>
skip_run = yes
...
</plot>

# data input
chr1 100 200 0.25 # read in
chr1 200 300 0.1  # read in
chr1 300 400 0.1  # not read in
chr1 400 500 0.1  # not read in
chr1 500 600 0.1  # not read in
chr1 600 700 0.3  # read in

```




使用min_value_change参数设置第n个值和第n-1个值(被读取绘制)绝对差异需大于min_value_change值。

```

<plot>
min_value_change = 5
...
</plot>

# data input
chr1 100 200 1 # read in
chr1 200 300 2 # not read in (difference = 1 < 5)
chr1 300 400 5 # not read in (difference = 4 < 5)
chr1 400 500 6 # read in      (difference = 5 >= 5)
chr1 500 600 3 # not read in (difference = 3 < 5)
chr1 600 700 13 # read in     (difference = 7 >= 5)
chr1

```



若没有使用min/max设置axis范围，那么axis将会跨越整个数据区域。同时可明确设置axis范围：min=-1，max=0，那么超过该范围的值将被隐藏。例如，使用rule隐藏value小于0的数据范围：

```

<rules>
<rule>
condition  = var(value) < 0
show      = no
</rule>
</rules>

```



axis方向默认为y-axis方向朝外，也就是较小的值比较大的值更靠近圆心。针对histogram，阳性bins值朝外，阴性bins值朝内。可设置y-axis方向：

`orientation=in`，将使得y-axis方向朝内，较大值更接近中心。

使用`fill_color=red`，对histogram进行颜色填充

合并两个histograms，一个正值一个负值，不同的背景颜色，效果更好。

Tiles

Tiles用于展示基因组区间(genes, exons, duplications)的跨度或覆盖度单元(clones, sequences, reads)。Tiles避免重叠会采用堆积的方式。

```

r1 = 0.98r
r0 = 0.86r

layers      = 15
margin      = 0.02u
orientation = out

thickness   = 15
padding     = 8

```

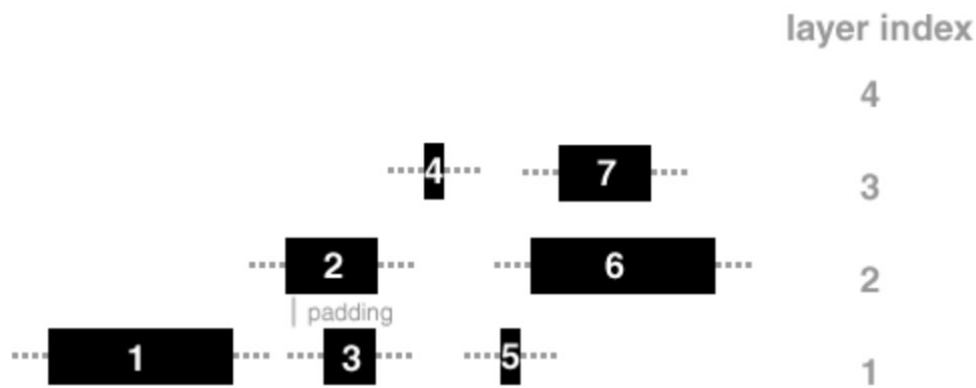
图形定义半径范围 $[0.86r, 0.98r]$ ，该值为相对于inner ideogram半径； $r0/r1$ 值分别对应tiles堆积的范围；

`orientation=out`，tiles将从 $r0$ 到 $r1$ 堆积，反之从 $r1$ 到 $r0$ ；如果`orientation=center`，tiles将从 $r0/r1$ 的中点向两边堆积；



图中每个tile单元半径宽度为15pixels，相邻单元具有8pixels的padding；

不同layers之间的空间为padding



假如tiles超过了layers数目(layers=15), layers_overflow参数将决定如何处理overflow的tiles:

layers_overflow=hide, 那么overflow(容纳不下的tiles)将会隐藏

layers_overflow=collapse, overflow的tiles会绘制于第一层

layers_overflow=grow, 有重叠, 增加新层, 不考虑层数限制

同时也可以采用rules模块

```
<rules>
<rule>
condition = var(size) > 100kb
color     = red
</rule>

<rule>
condition = var(size) > 50kb
color     = orange
</rule>

<rule>
condition = var(size) > 20kb
color     = yellow
</rule>
```

Heat Maps

热图用于高亮 2D图形, 在热图中, 根据基因区域的值显示对应的颜色。不同于其他图形, heatmap需要一个颜色列表。该颜色列表表明可明确定义在参数中, 或使用提前定义在colors模块中的颜色信息。

```
<plot>
type    = heatmap
file    = data/6/snp.number.1mb.txt
# color list
color   = spectral-5-div
r1      = 0.95r
r0      = 0.80r
</plot>
```

由于lines plots, histograms和heatmaps都使用相同的数据格式, 因此只需改变type值便可改变图像类型。

如果min和max值未定义, 那么数据点点颜色会根据值进行分配:

```
color_index = num_colors * (value-min)/(max-min)
```

如果定义了一个或全部min/max值，那么将会在以上公式中使用。

Heatmaps使用提前定义的颜色列表，可以使用具体的颜色列表名称来指定颜色使用：spectral-11-div 包含11-color spectral diverging brewer palette

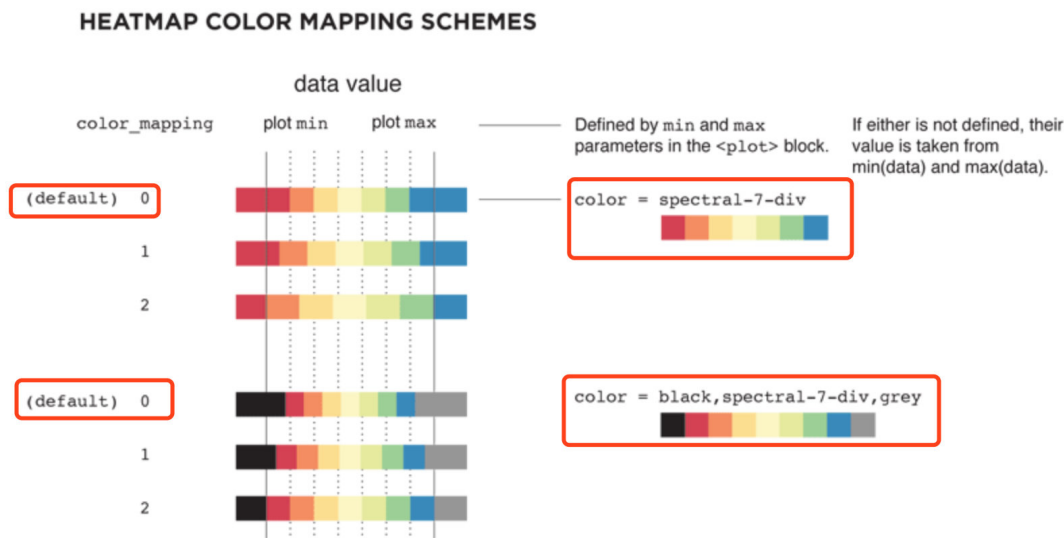
```
color=spectral-11-div
```

同样可以合并颜色列表扩大颜色范围：

```
color=ylorrd-9-seq-rev, ylgnbu-9-seq
```

```
color=black,ylorrd-9-seq-rev,white,ylgnbu-9-seq,grey
```

默认条件下，color_mapping=0：



Text Labels

文本标签用于描述一个基因的跨度信息，同散点图一样，text也会出现在区域的中间。数据格式如下：

```
...
hs1 225817866 225910748 ZNF678
hs1 26560711 26571853 ZNF683
hs1 40769819 40786426 ZNF684 color=red
hs1 149521414 149531004 ZNF687
...
```

基本的文本标签参数如下，需放置在多重plots中：

```

<plots>

<plot>
type          = text
color         = black
file          = data/6/text.genes.znf.txt

r0 = 0.4r
r1 = 0.8r

show_links    = yes
link_dims     = 4p,4p,8p,4p,4p
link_thickness = 2p
link_color    = red

label_size    = 24p
label_font    = condensed

padding = 0p
rpadding = 0p

</plot>

</plots>

```

文本标签默认值位于etc/tracks/text.conf中，取消默认值可采用：

```
color=undef
```

或取消所有默认值，undefind track_defaults

```

<<include etc/housekeeping.conf>>
track_defaults* = undef

```

标签内容限制在r0/r1范围内，如果使用很窄范围定义很大的标签，该标签信息不会显示：

```

r0          = 1.05r
r1          = 1.1r
label_size = 100p

```

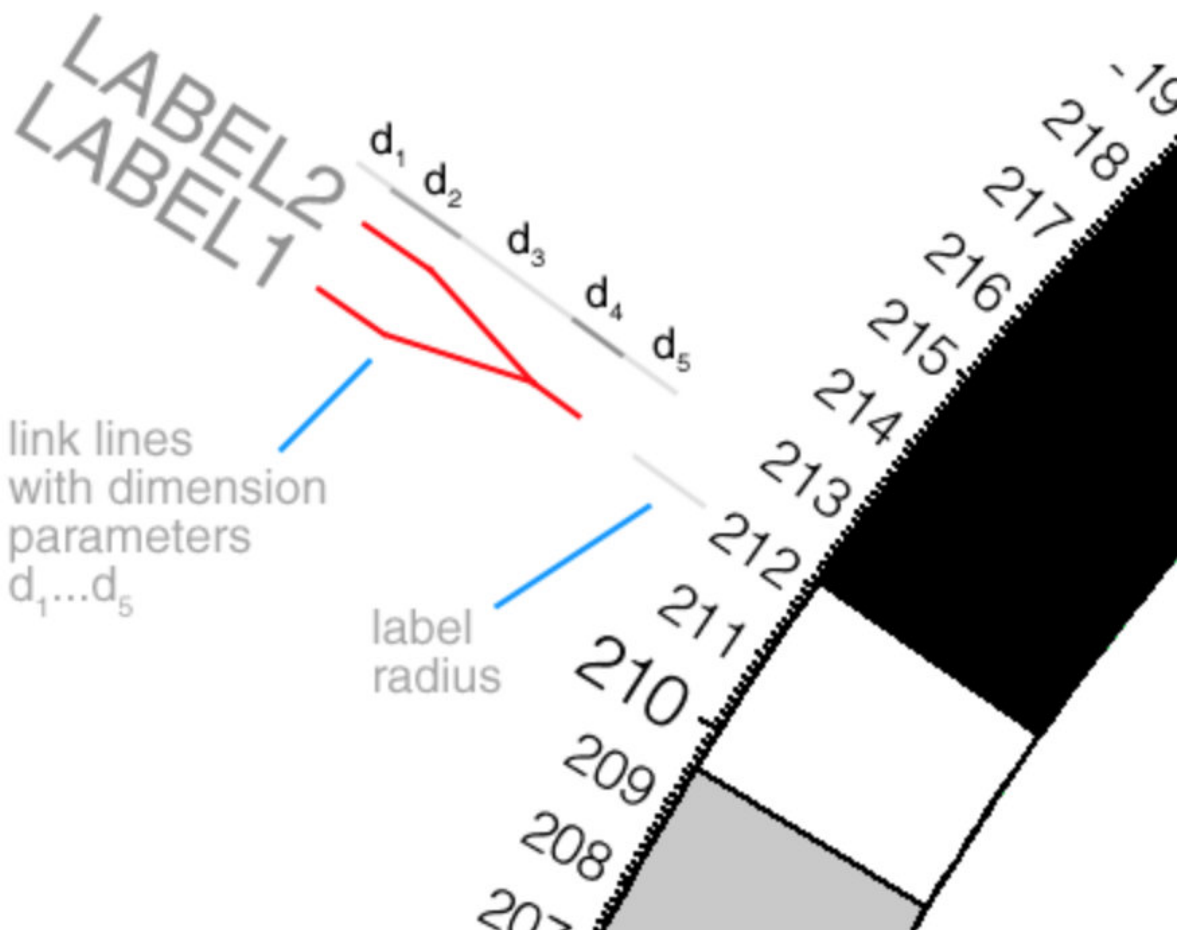
最简单设置为，扩大r1值：

```

r0          = 1.05r
r1          = 1.05r+300p
label_size = 100p

```

由于重叠的标签在图中通过偏移避免(开启snuggling参数)，link_dims参数指定了连接标签和基因组位置的连线的dimensions，该连线尤为重要，每个连线拥有5个dimensions信息: outer padding, outer line length(drawn at new position), connecting line length(connects old to new position), inner line length(drawn at old position), inner padding:

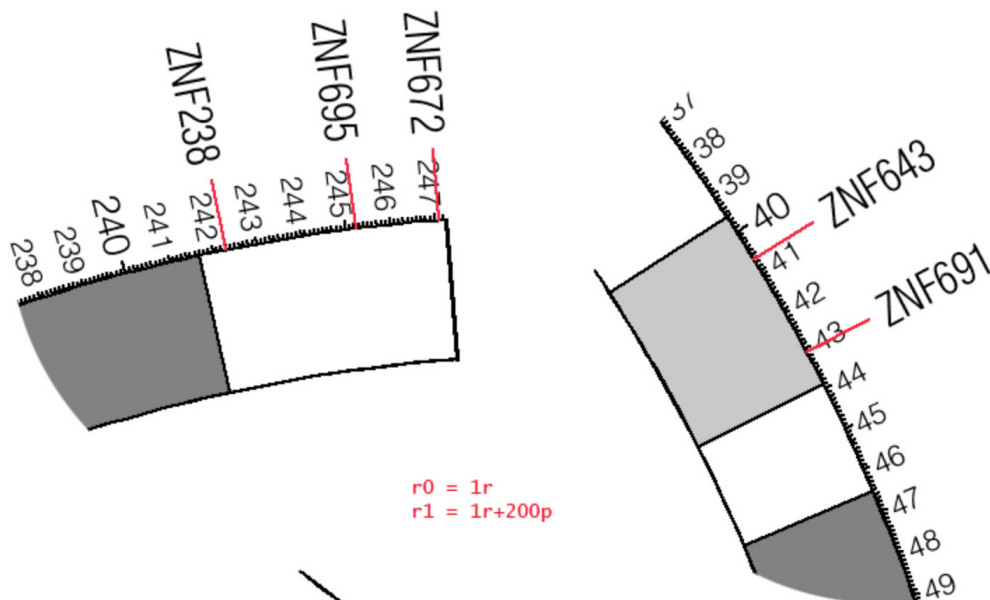


默认条件下，标签朝外，连线指向圆，同时标签内容靠左排列

选择label_rotate=no(默认条件下，标签将会旋转适应其基线方向)，如果标签很短，可选择label_rotate=no，取消旋转。

```
<plot>
...
r0 = 1r
r1 = 1r+200p
show_links      = yes
link_dims       = 0p,0,50p,0p,10p
link_thickness  = 2p
link_color      = red
</plot>
```

通过设置标签半径 $r_0=r_1$ ，标签将会绘制在刻度上。



Text Stacking

默认条件下标签会通过stack避免重叠出现，也就是出现重叠时会通过平移避免重叠 (label_snuggle=yes)，可通过增加max_snuggle_distance使得标签的分布更加紧凑，当采用snuggling时，可调整padding值有助于snuggling。angular(via padding)和radial(via rpadding)的方向都是围绕标签，可以是绝对值或相对值，用于控制文字边缘大小。

```

<plot>

...
label_snuggle           = yes
# shift label up to 2x its height in pixels in the angular direction
max_snuggle_distance    = 2r

# sample possible label positions every 2 pixels
snuggle_sampling        = 2

snuggle_tolerance       = 0.25r

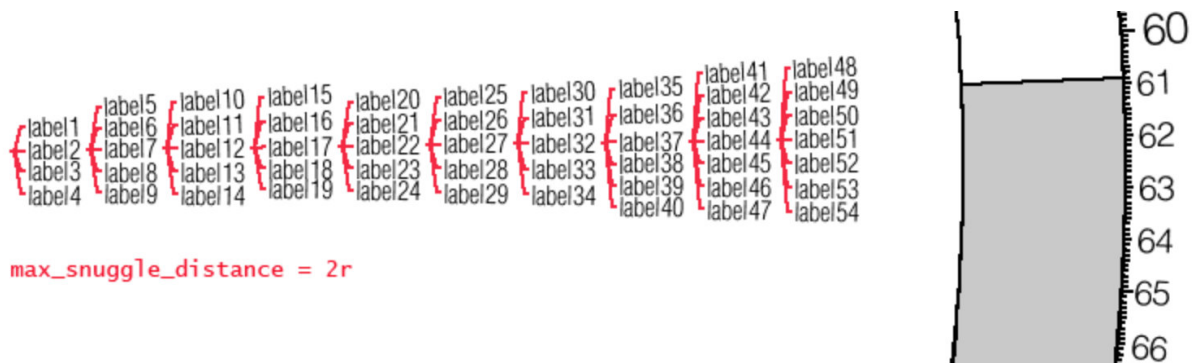
snuggle_link_overlap_test = yes
snuggle_link_overlap_tolerance = 2p

snuggle_refine          = yes

</plot>

```

Max_snuggle_distance决定了并排放置标签数目



若设置padding为负值，那么标签将更紧密显示。radial padding为相对于标签的宽度，angular padding为相对于标签高度。

```
<plot>
...
padding = 2p
rpadding = 0.1r

</plot>
```

若不想使用snuggling, text将一致排列放置。若将text在一个定义的区域(sequence)内不采用snuggling处理, 可以关闭(默认就是关闭), 使用monospaced font(等宽字体):

```
label_font=mono
```

所有字体都定义在etc/fonts.conf内

```
# etc/fonts.conf


...
mono          = fonts/modern/cmuntt.ttf # CMUTypewriter-Regular
mono_light    = fonts/modern/cmunbtl.otf # CMUTypewriter-Light

# same as mono and mono_light
fixed         = fonts/modern/cmuntt.ttf # CMUTypewriter-Regular
fixed_light   = fonts/modern/cmunbtl.otf # CMUTypewriter-Light
...
```

GLYPHS-PART I

rules可用于通过value参数调整标签文本, 例如所有标签均显示为X:

```
<rule>
condition = 1
value     = X
</rule>
```



同时可以合并多个rule, 例如可根据文本字符改变现实颜色, 同时该rule必须有flow=continue来允许接下来的其他rule作用, 否则会完成当前后会终止:

```
# first change color
<rule>
condition = var(value) eq "A"
color     = red
flow      = continue
</rule>

<rule>
condition = var(value) eq "T"
color     = blue
flow      = continue
</rule>

<rule>
condition = var(value) eq "C"
color     = green
flow      = continue
</rule>

# change label text, for all data points
<rule>
condition = 1
value     = X
</rule>
```

同时可以通过修改font, 调整文本标签为想要的glyph(图形符号)。例如, 使用符号字符(etc/fonts.conf, glyph, fonts/symbols/symbols.ttf)获得圆形符号:

```

<plot>
label_font = glyph

<rules>
...
<rule>
condition = 1
value     = m
</rule>
...
</rules>

```

符号字体定义如下：

```

small
| medium
| | large
| | |
a b c   square
d e f   rhombus
g h i   triangle up
j k l   triangle down
m n o   circle

upper case - solid
lower case - hollow

```

GLYPHS-PART II

在绘制图形轨迹时常需要将文本标签修改为符号标签

```

<rules>

flow = continue

<rule>
condition = var(value) =~ /cancer/
color     = red
</rule>

<rule>
condition = var(value) =~ /omim/
color     = green
</rule>

<rule>
condition = var(value) =~ /other/
color     = blue
</rule>

<rule>
condition = 1
value     = N
</rule>

</rules>

```

或通过rule隐藏其他基因类型：

```

<rule>
condition = var(value) !~ /cancer/
# hide anything that doesn't match "cancer"
show = no
</rule>

```

通过glyph的大小来表示数据点的density，由于circos不回计算density，需提前处理数据将density现实为label_size：


```

...
hs1 3000000 3000000 cancer label_size=1
hs1 6000000 6000000 cancer label_size=2
...
hs1 1000000 1000000 omim label_size=9
hs1 2000000 2000000 omim label_size=14
...
hs1 1000000 1000000 other label_size=26
hs1 2000000 2000000 other label_size=10
...

```

使用rule模块，将label_size重新映射到其他值，这里原始label_size范围为1p到42p：

```

# linear remap to [6,50]
label_size = eval(remap_int(var(label_size),1,42,6,50))

# ... with shallower increase
label_size = eval(remap_int(sqrt(var(label_size)),1,sqrt(42),6,50))

# ... with steeper increase
label_size = eval(remap_int(var(label_size)**2,1,42**2,6,50))

```

Connectors

connectors为ideogram上两半径位置的相关区域的连线

```

<plot>

type = connector
file = data/6/connectors.txt

r0    = 0.8925r
r1    = 0.999r

connector_dims = 0,0.3,0.4,0.3,0

thickness = 2
color     = black
</plot>

```

数据文件定义了用于连线的r0和r1位置

```

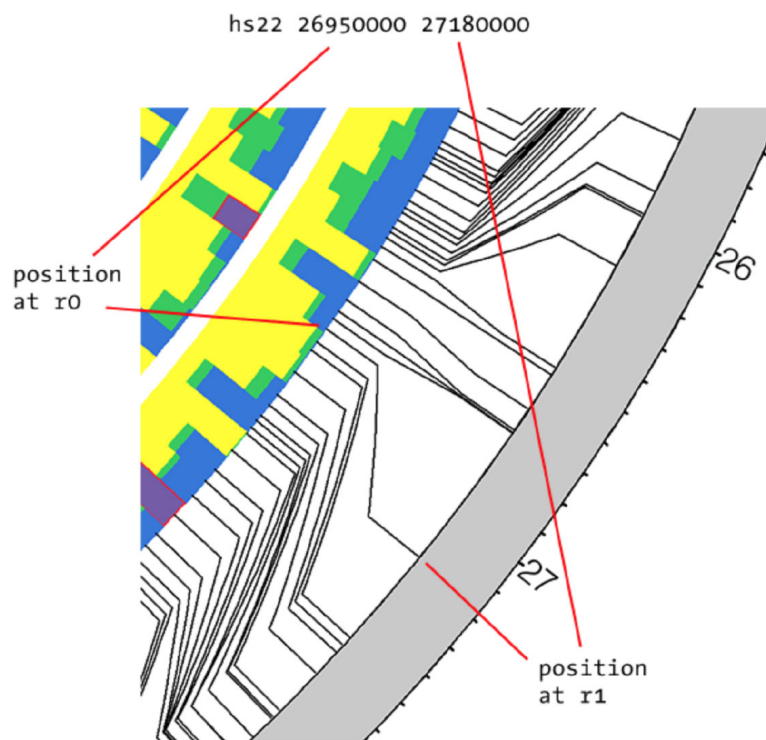
# connectors.txt
...
hs22 15116257 16046604
hs22 15136864 16056568
...

```

其中connector_dims信息如下：



r0和r1位置描述如下：



定义标准及必须参数

```

4 ##包含必要默认参数
5 <image>
6 <<include etc/image.conf>>
7 </image>
8 <<include etc/colors_fonts_patterns.conf>>
9 <<include etc/housekeeping.conf>>
10
11 # circos.conf
12 karyotype=practise_contig_cytoideogram.txt #染色体文件名
13 #chromosomes_reverse=/NODE_22/ #是否对染色体进行反转
14 chromosomes_units=10000 #染色体单位长度bp
15 dir=. #输出图片文件夹
16 svg=no #是否输出svg文件
17

```

定义ideogram

```

18 #####
19 <ideogram>
20 <spacing>
21 default=0.005r #染色体之间距离
22 </spacing>
23 radius=0.8r #染色体位置,1r表示默认分辨率1500pixels
24 thickness=35p #染色体厚度
25 #stroke_color=dgrey #染色体体轮廓颜色
26 #stroke_thickness=2p #染色体轮廓厚度
27 fill=yes #是否对染色体颜色填充
28 show_label=yes #是否显示标签
29 label_font=default #染色体标签字体
30 label_radius=1r+100p #相对于ideogram的半径
31 label_size=20 #标签字体大小
32 label_parallel=no #和染色体垂直, yes表示染色体标签方向同染色体方向, 易于查看
33 </ideogram>
34 #####

```

添加tick配置

```

1 ##ideogram:chromosomes_units=10000
2 show_ticks=yes #显示刻度
3 show_tick_labels=yes #显示刻度标签
4 <ticks>
5 #整体ticks参数
6 radius=1r #显示刻度相对于ideogram的半径位置
7 #radius=dims(ideogram,radius_outer)+25p #使用dims()指定feature+feature parameter, 这里绘制在外层
8 color=black #刻度线颜色
9 thickness=2p #刻度线厚度
10 multiplier=1e-4 #显示对应刻度值: 实际spacing乘以该值
11
12 <tick>
13 #orientation=out #刻度朝向, 默认out
14 #绝对刻度距离可以定义为碱基单位,b;或这染色体单位,u,由chromosomes_units定义
15 spacing=10u #每单位长度出现一刻度,对应chromosomes_units单位值
16 color=red #不继承ticks颜色定义, 定义刻度颜色为红色
17 show_label=yes #显示刻度标签, 标签值按照multiplier计算
18 label_color=red #显示标签颜色
19 size=18p #刻度线大小
20 #size=0.1r #等同于ideogram厚度的1/10
21 #thickness=0.1r #等同于刻度长度的1/10
22 label_size=25p #标签大小
23 label_offset=5p #刻度与标签之间距离
24 format=%d #标签数字格式, %d表整数, %f表浮点数
25 </tick>
26 <tick>
27 spacing=2u
28 show_label=yes
29 label_color=green
30 size=25p
31 label_size=15p
32 label_offset=5p
33 format=%d
34 </tick>
35 </ticks>

```

绘制gc skew线

```

1 ##line plot for gc skew
2 ##朝向, 默认正数朝外orientation=out, 其他数值相对该默认朝向显示
3 type=line #plot类型
4 thickness=2p #折线厚度
5
6 <plot>
7 file=38588_gc_positive.position #折线输入文件
8 color=red #折线颜色
9 max_gap=0u #相邻数值间间隔
10 r0=0.48r #折线内径, 相对于当前ideogram
11 r1=0.57r #折线外径, 相对于当前ideogram
12 orientation=out #折线防线朝外
13 </plot>
14
15 <plot>
16 file=38588_gc_negative.position
17 color=blue #折线颜色
18 max_gap=0u
19 r0=0.39r
20 r1=0.48r
21 orientation=out #折线防线朝外, 但是负值对应朝内
22 </plot>

```

显示文本内容

```

1 ## text plot configuration
2 <plot>
3 type=text #定义plot为文本标签
4 #file=38588_Contig_Gene.txt # text文本数据
5 file=card_38588.txt
6 color=red #定义标签颜色
7 label_font=condensed #定义标签字体
8 label_size=15p #定义标签大小
9 r0=0.7r #定义标签内径位置, 当r0=r1 (1r=1r) 时标签显示在染色体轴上
10 r1=0.95r #定义标签外径
11
12 label_snuggle=yes #避免多个标签过近重叠
13 max_snuggle_distance=2r #显示对应标签并排放置数
14 snuggle_refine=yes
15 show_links=yes #显示标签和染色体连接线
16 link_color=black #连接线颜色
17 link_thickness=2p #连接线厚度
18 link_dims=5p,10p,30p,20p,5p #各连线五部分长度
19
20 </plot>
21

```

对应文件显示轴线

```

37 <plots>
38 <<include line_38588.conf>> #使用line_38588.conf作为折线图参数
39 <<include text_38588.conf>> #使用text_38588.conf作为文本图参数
40 <axes>
41 show=data #仅含以上数据处显示, 因此包含在其他plot中
42 #<axis>
43 #spacing=0.1r #每个plot范围内, 间隔0.1r绘制一条axis
44 #color=black #axis颜色
45 #</axis>
46 <axis>
47 position=1r #axis位置, 现对于以上数据,折线图的位置
48 color=green #axis颜色
49 </axis>
50 <axis>
51 position=0r #其他每一个图形的r0/r1范围内, <<include line_38588.conf>>内含两个图,因此每图绘制一条
52 color=black #axis颜色
53 thickness=2
54 </axis>
55 </axes>
56 </plots>
57 #####

```