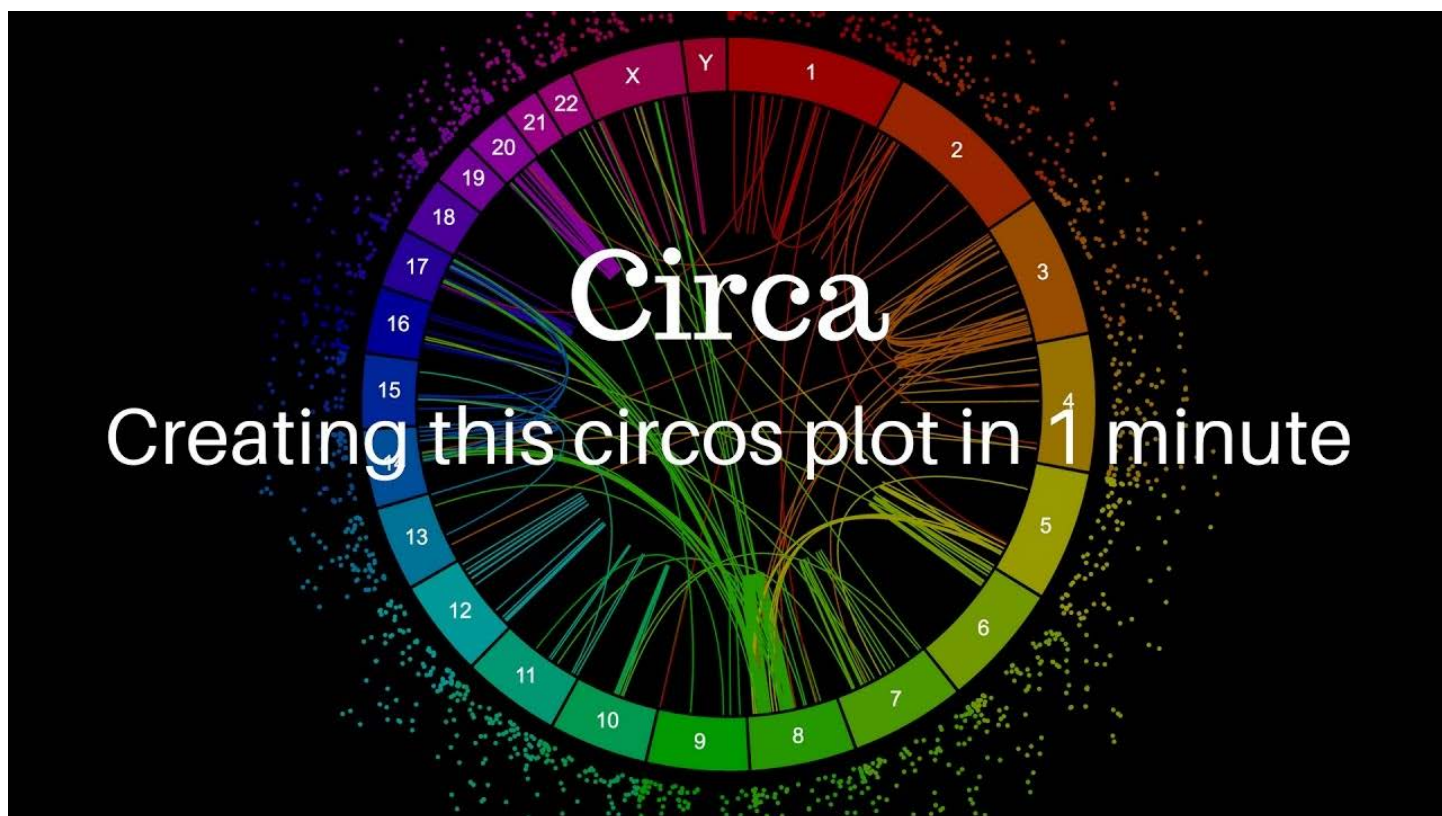


Circos配置文件



Circos的配置文件准备

Circos的使用主要通过输入一个配置文件，改配置文件的主要内容以各种区块表示，大区块中可以包含小区块。区块中以”变量=值”的方式来进行参数的设定：

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

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

有些配置文件不需要改动，比如颜色、字体等，一般将这类信息保存到一个独立的配置文件中。只需要在主配置文件中声明包含这些独立的配置文件名及其路径，即表示使用这些配置信息。最常用的放置到主配置文件尾部的数行

设置生成的图片参数

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

设置颜色、字体、填充模式的配置信息、

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

系统与debug参数

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

Circos的使用参数

```
-version
查询Circos版本
-modules
检测perl模块
-conf <string>
输入主文件配置
-outputdir <string>
设置输出文件名，该参数的值以.png为后缀
-svg
生成svg结果文件
-nosvg
不生成svg结果文件
```

Circos配置文件详解

ideogram block显示染色体

将染色体在圈图上展示出来，代表每个染色体的图形，称为ideogram。将以下配置信息放入一个单独的配置文件中，命名为ideogram.conf

```
<ideogram>
##设定ideogram之间的间隙
<spacing>
#设置圈图染色体之间的间隙大小，以下设置为每个间隙大小为周长的0.5%
```

```
default = 0.005r
#也可以设置指定的两条染色体之间的间隙
#<pairwise chrA01;chrA02>
#以下设定为两条染色体之间的间隙为圆的20度角
#spacing = 20r
#</pairwise>
</spacing>
##设定ideogram
#设定ideogram的位置，一下设定为ideograms在图离圆心的90%处
radius = 0.90r
#设定ideograms是否填充颜色。填充的颜色取决于karyotype指定的文件的最后一列
fill = yes
#设定ideograms轮廓的颜色及其厚度。如果没有该参数或设定其厚度为0，则表示没有轮廓
stroke_color = dgrey
stroke_thickness = 2p
## 设定label的显示信息
#设定是否显示label，label对应着karyotype文件的第四列，如果其值为yes，则必须要有label_radius参数来设定label的位置，否则会报错无法生成结果
show_label = yes
# 设定label的字体
label_font = default
# 设定label字体大小
label_size = 40
# 设定label的字体方向，yes是易于浏览的方向
label_parallel = yes
</ideogram>
```

ticks block以刻度形式显示染色体大小

将染色体的大小以刻度的形式在圈图上展示出来，将以下配置信息放入到一个独立的配置文件中，命名为ticks.conf

```
#是否显示ticks
show_ticks = yes
#是否显示ticks的labels
show_ticks_labels = yes
#设定ticks
<ticks>
##ticks的设置
#设定ticks的位置
radius = 1r
#设置ticks的颜色
color = black
#设置ticks的厚度
thickness = 2p
#设置ticks'label'的值的计算，将改刻度对应的值*multiplier得到展示到圈图上的label值
multiplier = 1e-6
#label值的格式化方法，%d表示结果为整数；%f表示结果为浮点数；%.nf表示结果保留小数点后n位
format = %d
## 以下设置了2个ticks，前者是小刻度，后者是大刻度
<tick>
#设置每个刻度代表的长度，若其单位为u，则必须要设置chromosome_units参数，比如设置chromosome_units=1000000,则如下5u表示每个刻度代表5M长度的基因组序列
spacing = 5u
```

```

#设置tick的长度
size = 10p
</tick>
<tick>
spacing = 25u
size = 15p
#由于设置的是大刻度，以下用于设置展示ticks'label
show_label = yes
#设置ticks'label字体大小
label_size= 20p
#设置ticks'label离ticks的距离
label_offset = 10p
format = %d
</tick>
</ticks>

```

links block以曲线连接显示基因组内部之间的联系

基因组内部不同的序列区域之间有联系，将之使用线条进行连接，从而展示到圈图上，常见的是重复序列之间的连接，将以下配置信息放入到一个独立的配置文件中，命名为links.conf

```

<links>
<link>
#指定link文件的路径，其文件格式为：
# chr1 start end chr2 start end
# chrA01 1 10 chrC01 10 45
#表明这两条染色体区域有联系，例如这个区域的序列长度>1kb且序列相似性>=90%
file = data/segdup.txt
#设置link曲线的半径
radius = 0.8r
#设置贝塞尔曲线半径，该值越大曲线越扁平，不好看
bezier_radius = 0r
#设置link曲线的颜色
color = black_a4
#设置link曲线的厚度
thickness = 2
<rules>
#以下可以设置多个rule，用来对link文件的每一行进行过滤或展示进行设定。每个rule都有一个condition参数，如果该condition为真，执行该block的内容，除非flow=continue，则不执行，检测下一个rule
#下面如果link文件中该行数据是染色体内部的link，则不进行展示
<rule>
condition = var(intrachr)
show = no
</rule>
#设置link曲线的颜色与ideogram的颜色一致，否则为统一的颜色
<rule>
#若condition为真，则执行该block的内容
condition = 1
#设置link曲线的颜色为第二条染色体的颜色，对应着link文件中第四列数据对应的染色体的名称
color = eval(var(chrC02))
#虽然condition为真，但依然检测下一个rule
flow = continue
</rule>
#如果link起始于chrA01，则其link曲线半径为0.99r
<rule>

```

```

contidition = from(chrA01)
radius1 = 0.99r
</rule>
#如果link结束于chrC01, 则其link曲线半径为0.99r
<rule>
condition = to(chrC01)
radius2 = 0.99r
</rule>
</rules>
</link>
</links>

```

plots block以直方图形式展示数据

将基因组序列中的GC含量、表达量等以直方图的形式在圈图中展示出来，将以下配置信息放入到一个单独的配置文件中，命名为plots_histogram.conf，下面绘制了两个直方图，并分别添加背景或网格线。

```

<plot>
#设定直方图
type = histogram
#数据文件路径, 4列
#chromosome start end data
# chrA01 0 2000000 155.0000
file = data/Bnal234.hist.txt
#设置直方图的位置, r1要比r0大, 直方图的方向默许为向外
r1 = 0.88r
r0 = 0.81r
#直方图的填充颜色
fill_color = vdgrey
#默认下直方图轮廓厚度为1px, 若不需要轮廓, 将其设置为0, 或者直接在etc/tracks/histogram.conf中修改
thickness = 0p
#直方图是由bins所构成的, 若bins在坐标轴上不相连, 最好设置不要将其bins连接到一起, 例如:
# chrA01 10 20 0.5
# chrA01 30 40 0.25
#上述数据设置值为yes和no时, 图形是不一样的
extend_bin = no
#以下添加rule, 不在chrA01上添加直方图
<rules>
<<include exclude.chrA01.rule>>
</rules>
#设置直方图的背景颜色
<backgrounds>
show = data
<background>
color = vvlgrey
</background>
<background>
color = vlgrey
y0 = 0.2r
y1 = 0.5r
</background>
<background>
color = lgrey
y0 = 0.5r
y1 = 0.8r
</background>

```

```

<background>
color = grey
y0 = 0.8r
</background>
</backgrounds>
</plot>
<plot>
type = histogram
#此处直方图的数据文件第4列是由多个由逗号分隔的数值，需要制作叠加的直方图
file = data/Bna.stacked.txt
r1 = 0.99r
r0 = 0.92r
#给4个值按顺序填充不同的颜色
fill_color = chrA01,chrA02,chrA03,chrA04
thickness = 0p
orientation = in
extend_bin = no
<rules>
<<include exclude.chrA01.rule>>
</rules>
#在直方图中添加坐标网格线
<axes>
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>
</plot>

```

plots block以热图形式显示数据

基因组一个区域内有多组数据时，适合以热图的形式显示数据，比如基因表达量，将以下配置信息放入一个单独的配置文件中，命名为plots_heatmap.conf

```

<plot>
#绘制heatmap
type = haetmap
#设置数据文件路径，文件有5列
# chrID start end data class
# chrA01 0 1999999 113.0000 id=chrA01
# chrA01 0 1999999 40.0000 id=chrA04

```

```

# chrA01 0 1999999 20.0000 id=chrA02
# chrA01 0 1999999 7.0000 id=chrA03
file = data/Bna.heatmap.txt
#设置图形所处位置
r1 = 0.89r
r0 = 0.88r
#设置热图的颜色, 颜色为chrA01, 以及相应带不同透明程度的5中颜色
color = chrA01_a5,chrA01_a4,chrA01_a3,chrA01_a2,chrA01_a1,chrA01
#设定scale_log_base参数, 计算颜色的方法如下:
#f=(value - min)/(max - min)热图中每个方块代表着一个值, 并给予相应的颜色标示, 一系列的值[min,max]对应一系列的颜色c[n],i=0..N
#n=N*f** (1/scale_log_base)
#若scale_log_base = 1, 则数值与颜色的变化是线性的
#若scale_log_base > 1, 则颜色向小方向靠近
#若scale_log_base < 1, 则颜色向大方向靠近
scale_log_base = 5
<rules>
<<include exclude.chrA01.rule>>
#仅显示id = chrA01的数据
<rule>
condition = var(id) ne "chrA01"
show = no
</rule>
</rules>
<plot>
type = heatmap
file = data/Bna.heatmap.txt
r1 = 0.90r
r0 = 0.89r
color = chrA02_a5,chrA02_a4,chrA02_a3,chrA02_a2,chrA02_a1,chrA02
scale_log_base = 5
<rules>
<<include exclude.chrA01.rule>>
<rule>
condition = var(id) ne "chrA02"
show = no
</rule>
</rules>
</plot>
<plot>
type = heatmap
file = data/Bna.heatmap.txt
r1 = 0.91r
r0 = 0.90r
color = chrA03_a5,chrA03_a4,chrA03_a3,chrA03_a2,chrA03_a1,chrA03
scale_log_base = 5
<rules>
<<include exclude.chrA01.rule>>
<rule>
condition = var(id) ne "chrA03"
show = no
</rule>
</rules>
</plot>
<plot>
type = heatmap
file = data/Bna.heatmap.txt
r1 = 0.92r

```

```

r0 = 0.91r
color = chrA04_a5,chrA04_a4,chrA04_a3,chrA04_a2,chrA04_a1,chrA04
scale_log_base = 5
<rules>
  <<include exclude.chrA01.rule>>
<rule>
  condition = var(id) ne "chrA04"
  show = no
</rule>
</rules>
</plot>

```

plots block以文本形式显示数据

若需要在圈图上显示基因名称，需要以文本形式展示，将以下配置信息放入到一个单独的配置文件中，命名为plots_text.conf

```

<plot>
#表示文字
type = text
#数据文件路径
file = data/Bnagene.labels.txt
#显示在图形中的位置
r1 = 0.8r
r0 = 0.6r
#标签的字体
label_font = light
#标签大小
label_size = 12p
#文字边缘的大小，设置较小则不同单词就可能会连在一块
#padding - text margin in angular direction
#rpadding - text margin in radial direction
rpadding = 5p
#设置是否需要在label前加一条线用来指出label的位置
show_links = no
link_dims = 0p,2p,5p,2p,2p
link_thickness = 2p
link_color = black
<rules>
  <<include exclude.chrA01.rule>>
#设置rule，对label中含有字母a或b的特异性显示
<rule>
  condition = var(value) =~ /a/i
  label_font = bold
  flow = continue
</rule>
<rule>
  condition = var(value) =~ /b/i
  color = blue
</rule>
</rules>
</plot>

```

rules block放置常用的规则配置

上述配置文件中，很多track没有在1号染色体上展示，需要设置如下规则信息，将之写入到文件exclude.chrA01.rule中

```
<rule>
condition = on(chrA01)
show = no
</rule>
```

主配置文件

在主配置文件circos.conf中，包含以上所有需要的配置文件信息，则可以会出所需要的track，此外，可以设置一些全局设置。

```
#指定染色体组型的文件，该文件有7列：
#chr - ID LABEL START END COLOR
#chr - chrA01 1 0 2154635789 chr1
#chr - chrA02 2 0 2451545665 chr2

karyotype = data/karyotype/karyotype.Bna.txt
#设置长度单位，以下设置表示1M长度的序列代表为1u
chromosomes_units = 1000000
#默认设置下是将karyotype文件中所有染色体都展示出来，当然也可以根据需要仅展示指定的染色体，使用如下的参数设置：

chromosomes_display_default = no
#以下参数设置指定的chromosome用于展示到圈图中，//中是一个正则表达式，匹配的chromosomes用于展示到圈图中，其匹配的对象是karyotype文件中的第3列，也可以直接列出需要展示的chromosomes，例如：chrA01;chrA02;chrA03;chrA04;chrA05

chromosomes = /chrA0[1-5]$/
#以下设置各个ideograms的大小，其总长度为1，chrA01的长度为0.5，chrA02,chrA03,chrA04这三条染色体的总长度为0.5，并且这三条染色体的长度是均匀分布的，注意前者的单位是r，后者使用了正则表达式匹配多条染色体，其单位于是为rn
chromosomes_scale = chrA01=0.5r,/chrA0[234]/=0.5rn
#使chrA02、chrA03和chrA04在圈图上展示的方向是相反的
chromosomes_reverse = /chrA0[234]/
#设置各个ideograms的颜色
chromosomes_color= chrA01=red,chrA02=orange,chrA03=green,chrA04=blue
#默认下在ideogram block中统一shezhileideograms的位置，可以使用此参数调整指定ideograms的位置
chromosomes_radius = chrA04:0.9r
#chromosomes_radius = chrA02:0.9r;chrA03:0.8r;chrA04:0.7r
#karyotype文件最后一行指定了各个chromosomes的颜色，而使用chromosomes_color参数也能修改颜色。使用如下方法进行颜色修改则更加直观，以下方式是对颜色重新定义。chr1,chr2,chr3,chr4对应着karyotype文件最后一行的值，代表着颜色的类型。此处使用color block来对其进行重新定义。注意重新定义的时候需要加符号*
<colors>
chr1* = red
chr2* = orange
chr3* = green
chr4* = blue
</colors>
###绘制plot图,将上述设置好的配置文件添加进来
<plots>
<<include plots_histogram.conf>>
<<include plots_heatmap.conf>>
<<include plots_text.conf>>
</plots>
<<include ideogram.conf>>
```

```
<<inlcude ticks.conf>>
<<include links.conf>>
#####
#插入必须的但不常修改的标准参数
<image>
<<include etc/image.conf>>
</image>
<<include etc/color_fonts_pattern.conf>>
<<include etc/housekeeping.conf>>
```

使用Circos命令绘图

先安装

```
##第一步先去官网下载最新版本
$ cd circos
$ tar zxf circos-tutorials-0.66.tgz
$ mv data data.bak
$ mv circos-tutorials-0.66/data/ ./
```

所有文件包括配置文件、数据文件等放置于当前目录文件夹中，运行circos绘图

```
$ ./bin/circos -conf circos.conf
```

这篇文章主要来自于[Circos的安装和简单使用](#),可以说是十分详细了。当然最好的文档还是官方提供的文档[Circos](#)。