Genome Visualization with Circos

Session 1 Preamble — Practical Sessions

Martin Krzywinski
Genome Sciences Centre
100-570 West 7th Ave
Vancouver BC V5Z 4S6 Canada
1-604-877-6000 x 673262
martink@bcgsc.ca
http://mkweb.bcgsc.ca

Circos
http://circos.ca

Course Materials
http://circos.ca/documentation/course

Genome Sciences Center
http://bcgsc.ca

Version History
v0.24 12 Jun 2017
v0.23 3 May 2016
v0.22 12 May 2014
v0.21 6 May 2014
v0.20 21 Apr 2014
v0.19 7 May 2012
v0.18 29 Jun 2011
v0.17 4 Jun 2011
v0.16 23 Jul 2010
v0.15 12 Jul 2010
v0.14 30 Jun 2010
v0.13 30 Jun 2010
v0.12 29 Jun 2010
v0.11 29 Jun 2010
v0.10 15 Jun 2010

CIRCOS.

round is good

CIRCOS @ mkweb.bcgsc.ca/circos

Table of Contents

Outline of Practical Sessions

During the practical sessions you will have an opportunity to work with Circos—each session will focus on a different aspect of the software.

Each session is divided into multiple lessons. Each lesson is a self-contained example during which you can explore specific aspects of the configuration file. Lessons build on one another—the configuration file of the previous lesson is used as a template for the next lesson.

SESSION 1 — DRAWING IDEOGRAMS

Drawing ideograms, and adjusting their spacing, size, order and scale.

SESSION 2 — DRAWING DATA

Adding data tracks, including histograms, heat maps and links.

You'll be introduced to dynamic rules in this session, which allow you to change the way your data is displayed based on their values.

SESSION 3 — DRAWING LINKS AND USING RULES

Advanced features, including bundling links and using dynamic rules to change the way data is displayed.

SESSION 4 — SELF STUDY: YEAST GENOME COMPARISON

You'll use Circos to visualize genome yeast duplications and conservation generated during Fredj's practical session during the *Pasteur Institute's Genome Analysis Course*.

SESSION 5 — CHALLENGE

Challenge in which you start with a configuration file I started for you and edit it to create a full image.

SESSION 6 — DRAWING DATA ON ONE LEISHMANIA GENOME

Drawing data tracks line line plots, heatmaps, tiles and text on one genome.

SESSION 7 — COMPARISON OF LEISHMANIA GENOMES

Drawing data tracks and links on two genomes.

Course Files

These instructions are specific to the workstation installation at the EMBO Bioinformatics and Genome Analysis course.

http://events.embo.org/coming-soon/index.php?EventID=pc17-07

https://webext.pasteur.fr/tekaia/BGA_courses.html

The handouts and slides are updated a few days in advance of the presentation. The contents here are for the course in 2017 in Thessalonica.

Basic command-line Linux skills are assumed.

LOAD CENTOS MODULES

First see whether perl and Circos are available

```
> which perl
/BGA2017/perl-5.24.1/bin/perl
> which circos
/BGA2017/circos-0.69-5/bin/circos
```

If `which` does not return anything, you'll need to load the module.

```
> module use /BGA2017/modulefiles
> module load circos
```

INSTALL COURSE MATERIALS

To install the lessons, `untar` the course archive. I'll assume that you will do so in your home directory.

```
# switch to your home directory
> cd ~
> tar xvfz /BGA2017/Course_material/June08/circos-course.tgz
# this might take a minute
```

COURSE MATERIALS FILE ORGANIZATION

The structure of the session files is as follows

```
circos-course/
  pov-collection.pdf   Collection of Points of View column on data visualization
  slides/   lecture slides
  handouts/ DOCX/PDF handouts (detailed explanations of each lesson)
  session/
    1/      Session 1 — drawing ideograms
      1/      Lesson 1
       etc/    configuration
      2/      Lesson 2
    ...
    2/      Session 2 — drawing data
    3/      Session 3 — drawing links and rules
    4/      Session 4 — yeast genome comparison
    5/      Session 5 — Circos challenge
    6/      Session 6 — Leishmania genome
    7/      Session 7 — Leishmania genome comparison
```

We have time to cover 4 practical sessions in the course materials. We'll use `session/6` and `session/7` for the first two practical sessions. The last session will use `session/4`, which includes data from Fredj's comparative genome lectures.

Using Course Files

During each session, you will be working entirely within the corresponding session directory.

I will assume that you have the course materials in `~/circos-course`.

For example, for Session 1, you will be working from `~/circos-course/session/6`.

```
> cd circos-course/session/6
> ls
# lesson files for this session
drwxr-xr-x. 12 bga2017 bga2017  101 Apr 27  2016 .
drwxr-xr-x. 12 bga2017 bga2017 4096 Apr 27  2016 ..
drwxr-xr-x.  3 bga2017 bga2017   52 Apr 27  2016 1
drwxr-xr-x.  3 bga2017 bga2017   33 Sep 26  2014 2
drwxr-xr-x.  3 bga2017 bga2017   52 Apr 27  2016 3
drwxr-xr-x.  3 bga2017 bga2017   52 Apr 27  2016 4
drwxr-xr-x.  3 bga2017 bga2017   52 Apr 27  2016 5
drwxr-xr-x.  3 bga2017 bga2017   52 Apr 27  2016 6
drwxr-xr-x.  3 bga2017 bga2017   52 Apr 27  2016 7
drwxr-xr-x.  3 bga2017 bga2017   33 Sep 26  2014 8
# shared data for this session
drwxr-xr-x.  3 bga2017 bga2017 4096 Oct  6  2014 data
# shared configuration files for this session
drwxr-xr-x.  2 bga2017 bga2017   23 Apr 27  2016 etc
# batch file that generates all images for this session
-rwxr-xr-x.  1 bga2017 bga2017   74 Sep 17  2014 run
```

Let's look at the contents of the first lesson.

```
> cd 1
> ls
# image created in this lesson
-rw-r--r--.  1 bga2017 bga2017 329465 Jun  7 11:31 circos.png
# configuration files for this lesson

# once you are done with Lesson 1, move to lesson 2
> cd ../2
```

Circos Configuration Files

Within each lesson, you will see a `etc/` directory. In this directory you will find one or more configuration files that are used to generate images for the lesson. A lesson will always have an `etc/circos.conf` file and it is this file that is used directly by Circos.

Files are included using the `<<include>>` directive. The included file is defined relative to the configuration file in which the `<<include>>` directive is found.

This configuration file will import contents from other files, such as (a) configuration files shared by other lessons in this session, (b) configuration files shared by all sessions in this course and (c) predefined configuration files in the Circos distribution that define default parameters.

```
karyotype = ../data/karyotype.5chr.txt

# this is a comment
chromosomes_units          = 1000000
chromosomes_display_default = yes

# ideogram position and labeling, specific to this lesson
<<include ideogram.conf>>
# tick spacing and position, shared by this session
<<include ../etc/ticks.conf>>
# image settings, shared by all sessions
<<include ../../../etc/image.conf>>

# remaining files are from the Circos distribution
# includes colors, fonts and pattern definitions
<<include etc/colors_fonts_patterns.conf>>
# important system settings
<<include etc/housekeeping.conf>>
```

Configuration files that are shared by all lessons in a session are found in `circos-course /session/NNN/etc`.

Configuration files that are shared by all sessions are found in `circos-course /session/etc`. For example `circos-course/session/etc/image.conf` defines image settings used throughout the course.

```
session/
       etc/       # shared by all sessions and all lessons
       1/
        etc/      # configuration shared by all lessons in session 1
          1/
            etc/  # configuration for session 2 lesson 1
          ...
```

Lesson Structure

The structure of each lesson will be the same. Files for lessons are independent – what you did in the previous lesson does not affect the configuration file of the next lesson.

When we start the practical sessions, I will ask you to navigate to the lesson directory (see above). Next, you'll load the relevant configuration file for that lesson into a text editor (e.g. emacs, vim). Typically this will be `etc/circos.conf` but sometimes we'll be working with `etc/ideogram.conf` or `etc/ticks.conf`. It will be easier if you can have each file open in a separate editor buffer, or window.

Once you have the file loaded in your editor, I will discuss parts of the file that are relevant to the lesson. You'll notice that there will be parts of the file that are commented out (lines prefixed with #). As we proceed through the lesson, I'll ask you to uncomment these lines (and possibly comment others, see below) to see how the change affects the image.

Whenever a change to the configuration file will be made, I will ask you to run Circos to generate a new image based on the changed configuration (see below).

If we don't have time to cover all the lessons for a session, I encourage you to finish the lessons on your own.

Editing Configuration Files

Let's use Session 6 Lesson 1 as an introduction to Circos configuration files. The
`etc/circos.conf` configuration file for this lesson

```
# The order, name, label and size of each chromosome is defined in the
# karyotype file. Look at this karyotype file to understand the
# format.
#
#  chr - NAME LABEL START END COLOR
karyotype = ../data/lm.karyotype.txt

# Any parameter values with suffix "u" will be a multiple of this
# number (e.g. 1000 = 1kb). Useful in definitions of ticks and ranges.
chromosomes_units = 1000

# Position and size of ideograms
<<include ideogram.conf>>

# Ticks and tick labels
<<include ticks.conf>>

# Image size, location
<<include ../etc/image.conf>>

# General parameters like colors, fonts, patterns and system settings.
<<include etc/colors_fonts_patterns.conf>>
<<include etc/housekeeping.conf>>
```

During the lessons, you will be asked to edit the configuration file and create an image. By
repeating this, you'll learn how the configuration file syntax works. By including in the file
configuration items that we will be using, I've tried to help you avoid errors due to malformed
configuration. If you get an error while running Circos (see below), make sure that you have not
inadvertently added spurious content to the file.

If you define a parameter more than once, you'll see an error like this

```
   *** CIRCOS ERROR ***
   CONFIGURATION FILE ERROR

   Configuration parameter [chromosomes] has been defined more than once in the
   block shown above, and has been interpreted as a list. This is not allowed.
   Did you forget to comment out an old value of the parameter?

   If you are having trouble debugging this error, use this tutorial to learn how
   to use the debugging facility

      http://www.circos.ca/tutorials/lessons/configuration/debugging
```

Make use of the online tutorials to learn more about configuration files. See

http://circos.ca/documentation/tutorials/configuration/configuration_files/

Running Circos

To use the lesson files you will need to install Circos. To create an image during a lesson, execute the `circos` binary and use the `-conf` flag to specify the input configuration file.

```
# generate image for Session 6 Lesson 1
> cd ~/circos-course/session/6/1]
> circos
debuggroup summary 0.30s welcome to circos v0.69-5 2 May 2017 on Perl 5.024001
debuggroup summary 0.30s current working directory /home/bga2017/circos-
course/session/6/1
debuggroup summary 0.30s command /BGA2017/circos-0.69-5/bin/circos [no flags]
debuggroup summary 0.30s guessing configuration file
debuggroup summary 0.30s found conf file /home/bga2017/circos-
course/session/6/1/etc/circos.conf
debuggroup summary 0.57s debug will appear for these features: output,summary
debuggroup summary 0.57s bitmap output image ./circos.png
debuggroup summary 0.57s parsing karyotype and organizing ideograms
debuggroup summary 0.58s karyotype has 35 chromosomes of total size 32,108,776
debuggroup summary 0.60s applying global and local scaling
debuggroup summary 0.61s allocating image, colors and brushes
debuggroup summary 10.71s drawing 35 ideograms of total size 32,108,776
debuggroup summary 10.71s drawing highlights and ideograms
debuggroup output 11.84s generating output
debuggroup output 12.22s created PNG image ./circos.png (329 kb)Circos will guess
the location of the configuration file relative to your current directory.
```

If Circos cannot find the configuration file, you'll see this error

```
*** CIRCOS ERROR ***

CONFIGURATION FILE ERROR

Circos could not find the configuration file. To run Circos, you need to
specify this file using the -conf flag. The configuration file contains all
the parameters that define the image, including input files, image size,
formatting, etc.

If you do not use the -conf flag, Circos will attempt to look for a file
circos.conf in several reasonable places such as . etc/ ../etc
```

Common Errors

Watch out for these common errors when editing the configuration file. Circos can identify some errors and produce a detailed message.

MISSING CONFIGURATION FILE

If you run Circos without specifying the configuration file with `-conf` and Circos cannot locate the file (it tries in `etc/, ../etc` and a few other places) you'll see this error

```
*** CIRCOS ERROR ***

CONFIGURATION FILE ERROR

Circos could not find the configuration file. To run Circos, you need to
specify this file using the -conf flag. The configuration file contains all
the parameters that define the image, including input files, image size,
formatting, etc.

If you do not use the -conf flag, Circos will attempt to look for a file
circos.conf in several reasonable places such as . etc/ ../etc
```

You can see where Circos tried to look by using `-debug_group io`

```
> circos -debug_group io
```

UNBALANCED CONFIGURATION BLOCKS

Make sure that you do not forget to close a block.

```
<rules>
 <rule>
 </rule>

 <rule>
</rules>
```

The missing `</rule>` for the second `<rule>` block will cause a parsing error. Don't forget closing `</links>`, `</plots>` and `</rules>` tags.

REDUNDANT PARAMETER DEFINITIONS

In most cases you are not allowed to have multiple definitions of a parameter.

```
<plot>
..
color = black
..
color = red
..
</plot>
```

Circos will catch this and tell you what parameter is defined more than once.

```
CONFIGURATION FILE ERROR

  Configuration parameter [color] in parent block [plot] has been
  defined more than once in the block shown above, and has been interpreted as a
  list. This is not allowed. Did you forget to comment out an old value of the
  parameter?
```

MISSING `eval()`

In a rule, if you want the parameter to be evaluated, don't forget `eval()`.

```
<rule>
condition = 1
color = var(chr)
</rule>
```

This will set the color to "`var(chr)`" without evaluating it to the actual value of the chromosome. You want

```
color = eval(var(chr))
```

When Circos does not recognize a color name it will default to black.

Debugging Circos

When things don't behave as you expect, you can use Circos' debugging facilities to narrow down the problem.

CONFIGURATION DUMP

Using the –cdump flag you can obtain the configuration file data structure. The output will reflect the hierarchical nature of the configuration file.

```
plots => {
         color => 'black',
         max => 1,
         min => O,
         plot => [
                 {
                  file => '../data/both.cons.2e6.max.txt',
                  fill_color => 'spectral-5-div-3'
                 },
```

The output is large, and best combined with grep. For example, to search for all parameter that match cache,

```
> circos -cdump | grep cache
         color_cache_file => 'circos.colorlist',
         color_cache_static => 1,
```

DEBUG GROUPS

There is a large number of diagnostic reports that can be generated during image creation, named by the associated functionality. The reports can be accessed using –debug_group and combined as a list (e.g. io,conf,timer).

```
# reports about file location
circos -debug_group io
# configuration file parsing and substitution
circos -debug_group conf
# timings
circos -debug_group timer
# all reports (long)
circos -debug_group _all
```

For a list of report groups see

http://circos.ca/documentation/tutorials/configuration/debugging