Paper 007-2011

SAS® & Circos:

Creating Circular Visualizations of Tabular Data

Danni Bayn Ph.D., Capella University, Minneapolis, MN

ABSTRACT

Representing complex data and relationships in a meaningful and enticing way is a timeless, but increasingly important, challenge for analysts. Circos is a highly customizable Perl module that can be used to create circular representations of tabular data that allow for improved visualization and enhanced insight. Unfortunately, it requires a fair bit of patience and a solid grasp of Unix and Perl to install and run Circos from the provided tutorials. That is where SAS® comes in. Rather than go through the frustration of installing dependencies and manually modifying a slew of textual configuration files, the program presented here does all of it for you. With this program, SAS® users can take existing data files and quickly transform them into stunning circular visualizations.

The intended audience for this paper is SAS® developers with a working knowledge of SAS® and a basic understanding of Perl.

Code was developed with SAS® 9.1.3 running under Windows XP Professional and Circos version 0.52-2 under Perl v5.10. Final code was tested under SAS® 9.2.

INTRODUCTION

As the world becomes more and more digital, a growing problem that faces analysts is not how to get data, but how to best represent it. This is especially true for fields that are not only rich in data, but also have high degrees of complexity or large numbers of variables. Circos is a tool, written in Perl, which takes complex data sets and represents them in a circular layout so that relationships between the many different variables can be analyzed and studied. Initially, Circos was developed to help researchers map genomic data. However, since its inception, the number of different fields it has been used for has sky-rocketed. For example, in April of 2010, in *Wired* magazine, Circos was used to create a circular representation of the complex and, oftentimes confusing, relationships for the characters in the popular TV series, *Lost*.

Despite its versatility for visualizing different kinds of data, Circos has yet to be really implemented outside of genomics and high profile media pieces. This is due, in part, to the complexity involved with installing, configuring, and running the software. Circos is written in Perl, requires the installation of at least a dozen or so supplemental modules, and relies exclusively on text based configuration files. The installation instructions assume a high degree of familiarity with Perl and access to a Unix based system. In addition, most of the tutorials for Circos are geared towards genomics, so casual users must be prepared to wade through a lot of discipline specific terms (e.g. Ideograms and Karyotypes).

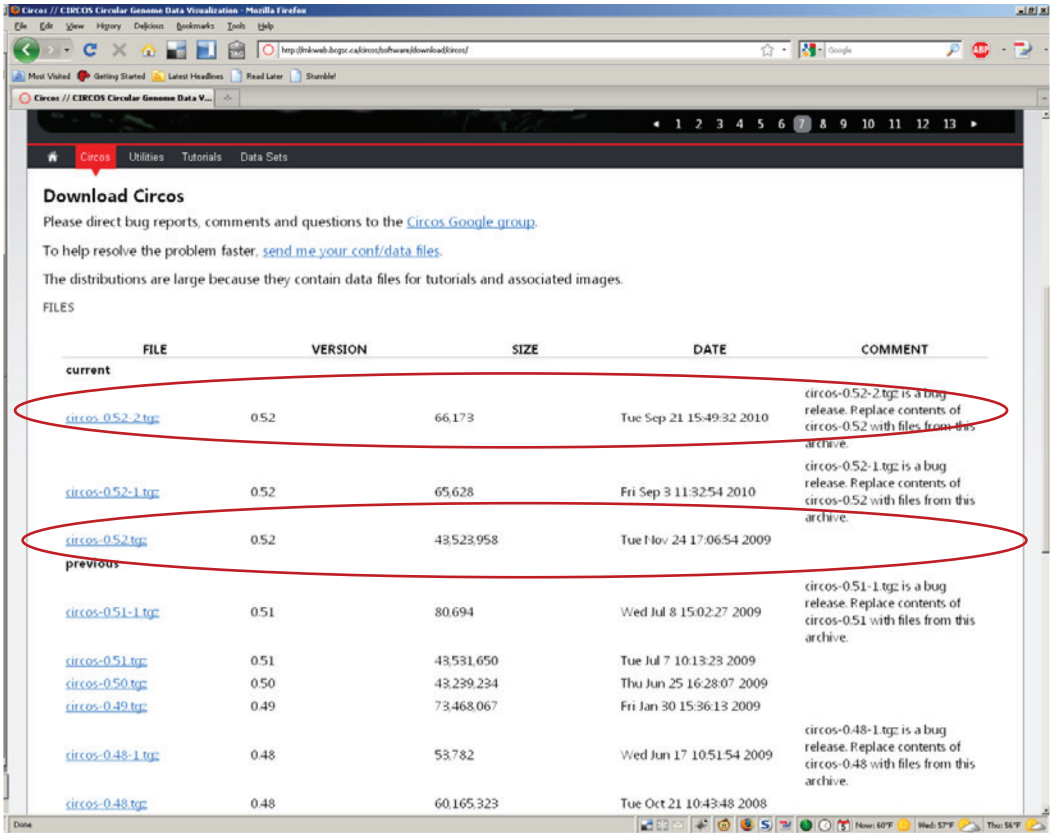
The purpose of this paper and the included program is to ease the process of installation and configuration, so that interested analysts can get a taste of what Circos has to offer, without having to become an expert in Perl or genomics first. In addition, the included program does not require users to have access to a Unix based system; instead, it is setup to run on a Windows machine.

GETTING STARTED

Before using the included SAS® program to build and configure Circos, you need to have Perl (v 5.10) installed on your system and the Circos source files extracted to a folder in your Perl directory.

First, to install Perl, go here: http://strawberryperl.com/ and select the installation package for your system. While you can tweak the installation settings if you would like, the default setup options are more than adequate to get you started. Depending on the speed of your system, this step should take between 5 and 15 minutes.

Once you have Perl installed, you need to download the Circos files. These files are available from http://mkweb.bcgsc.ca/circos/software/download/. You will need to download two files. The first is circos-0.52.tgz and the second is circos-0.52-2.tgz (Figure 1).


Circos // CIRCOS Circular Genome Data Visualization - Mozilla Firefox
File Edit View History Delicious Bookmarks Tools Help
http://mkweb.bcgsc.ca/circos/software/download/circos/
Most Visited Getting Started Latest Headlines Read Later Stumble!
Circos // CIRCOS Circular Genome Data V...
1 2 3 4 5 6 7 8 9 10 11 12 13
Circos Utilities Tutorials Data Sets
Done
Now: 60°F Wed: 57°F Thu: 56°F

Download Circos

Please direct bug reports, comments and questions to the Circos Google group.

To help resolve the problem faster, send me your conf/data files.

The distributions are large because they contain data files for tutorials and associated images.

FILES

FILE	VERSION	SIZE	DATE	COMMENT
current				
circos-0.52-2.tgz	0.52	66,173	Tue Sep 21 15:49:32 2010	circos-0.52-2.tgz is a bug release. Replace contents of circos-0.52 with files from this archive.
circos-0.52-1.tgz	0.52	65,628	Fri Sep 3 11:32:54 2010	circos-0.52-1.tgz is a bug release. Replace contents of circos-0.52 with files from this archive.
circos-0.52.tgz	0.52	43,523,958	Tue Nov 24 17:06:54 2009	
previous				
circos-0.51-1.tgz	0.51	80,694	Wed Jul 8 15:02:27 2009	circos-0.51-1.tgz is a bug release. Replace contents of circos-0.51 with files from this archive.
circos-0.51.tgz	0.51	43,531,650	Tue Jul 7 10:13:23 2009	
circos-0.50.tgz	0.50	43,239,234	Thu Jun 25 16:28:07 2009	
circos-0.49.tgz	0.49	73,468,067	Fri Jan 30 15:36:13 2009	
circos-0.48-1.tgz	0.48	53,782	Wed Jun 17 10:51:54 2009	circos-0.48-1.tgz is a bug release. Replace contents of circos-0.48 with files from this archive.
circos-0.48.tgz	0.48	60,165,323	Tue Oct 21 10:43:48 2008	

Figure 1: Download page for Circos. Download both of the circled files.

Using a program like WinZip (or any other folder compression/decompression utility), extract the files in circos-0.52 to your Perl folder. If you selected all of the default options when installing Perl from the previous step, the path should be C:\Strawberry\Perl. Extracting the files to this folder will create a new directory called circos-0.52.

Next, extract the Circos.pm file from the second file you downloaded, circos-0.52-2.tgz. Save this file to the lib folder within your new Circos directory. If you used the default options so far, the path will be C:\Strawberry\Perl\circos-0.52\lib. Overwrite the old Circos.pm file that is there.

That's it. The program will take care of the rest.

%INSTALL_CIRCOS

The %INSTALL_CIRCOS macro does three things:

1. It installs all Perl modules (that are not included in the default Perl installation) that Circos needs.
   Example. x "ppm install MinGW";
   x "ppm install clone";
2. It runs the Build files necessary to complete the Circos install.
   Example. x "cd C:\Strawberry\Perl\circos-0.52";
   x "perl build.pl";
3. It creates backup copies of all configuration files.
   Example. x "copy circos.conf circos.conf.bak ";
   x "copy make-table.conf make-table.conf.bak ";

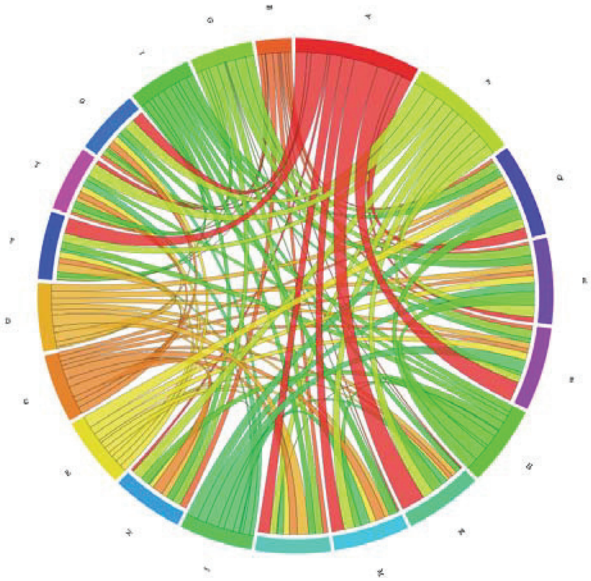
This macro has no parameters and will only need to be run once.

%SAMPLE_GRAPH

This macro creates a sample dataset and sample graph to ensure that your installation has completed successfully. If you did not go with the default options when installing Perl, make sure to adjust the CIRCOS_PATH macro variable at the beginning of the program to match your system. If you went with the default options in installing Perl above, this should not need to be changed. In addition, adjust the OUT macro variable, to be wherever you want the tables to be saved.

- %let CIRCOS_PATH = C:\strawberry\perl\circos-0.52\tools\tableviewer;
- %let OUT = H:\reporting\ad_hoc\Graph.svg;

After running %SAMPLE_GRAPH, you should have the following file in your OUT directory.



Note: Circos creates SVG files as its output. SVG files are Adobe standard Scalable Vector Graphics, and they can be viewed in any browser.

%CREATE_GRAPH

The %CREATE_GRAPH macro takes a data set name and, presuming the data is properly formatted, creates the input data set and the final circular graph.

Input data should look like this:

VAR1	Alabama	Alaska	Arizona	Arkansas
Alabama	-	2912	896	341
Alaska	1392	-	1904	332
Arizona	1028	2983	-	1289
Arkansas	1009	95	2872	-

Table 1: Sample data for Circos. State population movement.

Note: The default configuration options in Circos presume two things about the input data set.

1. Missing values are represented by a '-'. This is not the default option in SAS®, so the first line of this macro is:

   - `option missing ='-';`

   This option can be changed in the Customize_graph macro (discussed below) or by manually changing the configuration files.

2. Variable names should have NO spaces. The default configuration options for Circos interpret spaces as a delimiter. For the initial program to run successfully, make sure that all embedded spaces are removed or changed to an underscore. For future runs, this option can also be changed via the Customize_Graph macro or by manually changing the configuration files.

The dataset used in this paper contains population migration numbers for 2007-2008. The values in the first column of Table 1, show the state people went to, while the states along the column headers in the first row, represent where people came from. Running the %CREATE_GRAPH macro on this dataset created the following visualization:

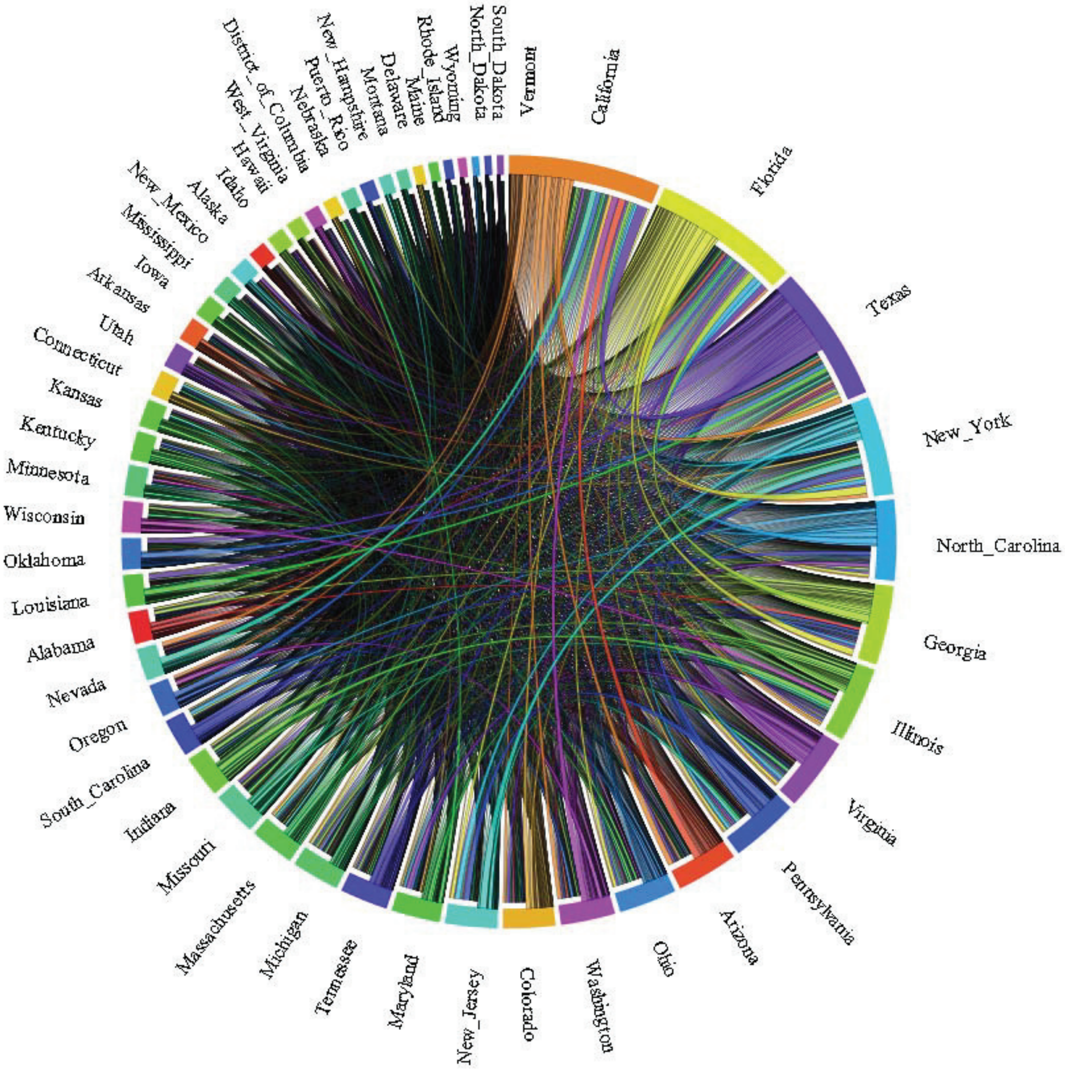


Figure 2: State to state migration for 2007-2008.

Now, as impressive as that looks, it is not really conducive to explaining how Circos works. So, the following figure was built using just the data in Table 1.

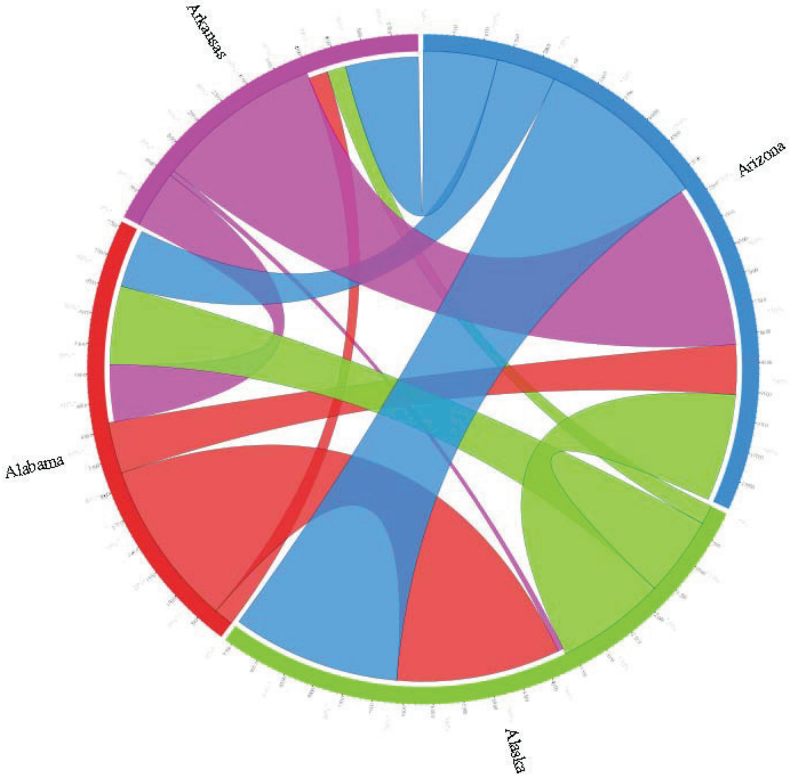


Figure 3: Subset of State to state migration for 2007-2008

Compared to Table 1, above, Figure 3 demonstrates how improved visualization of data can provide additional and improved insights. Row and column variables are distinguished by the relationship of the ribbon to the edge of the circle. A connected ribbon and edge indicates the row variable, while a space between the two represents the column variable (these can also be determined by color). For example, the wide blue ribbon traveling across the center of the circle represents the number of people that went from Arizona to Alaska (2,983), whereas, the green ribbon at the bottom right of the graph, represents the number of people that went from Alaska to Arizona (1,904).

%CUSTOMIZE_GRAPH

Parameter	Description
MOD_ORIG	Controls whether customizations should be made from the default values or build off of previous customizations.
FONT_SIZE	Size of the Labels used in the Graph.
RADIUS	Size of the Circular layout when compared to the Image size.
RIBBON_VAR	Determines if the end points of each "Ribbon" should be the same size.

The final macro provided is %Customize_Graph. Circos has six different configuration files that it uses to create visualizations like the ones shown above. In these configuration files, every little feature can be tweaked and modified. This macro picks some of the more useful modifications and provides a quick and easy method to change them. For example, in the default configuration, the "Ribbons" end points are the same width. By changing this parameter to 'Yes' the width of the endpoints is changed to reflect the actual values. In the State to State dataset above, this means that the Ribbon between Alaska and Alabama have different widths to represent the difference in population movement for people moving from Alaska to Alabama versus people moving from Alabama to Alaska.

The %Customize_Graph macro creates and executes a perl file that, based on the given parameters, modifies the relevant configuration files.

CONCLUSION

Despite having a bit of a steep learning curve, Circos is a robust and highly customizable Perl module for creating truly stunning visualizations. I strongly recommend that anyone who has had their curiosity piqued by the sample visualizations in this paper, visit their website and read through the tutorials they have to offer. While the terminology can be a bit hard to follow, the capabilities are truly amazing.

REFERENCES

Henderson, Don. (2009). http://www.SAS®community.org/wiki/Create_a_CSV_file_without_column_names/headers_in_row_1#Run_PROC_EXPORT_and_use_a_DATA_step_to_rewrite_the_file_without_the_first_row .

Krzywinski, M. et al. (2009). Circos: An Information Aesthetic for Comparative Genomics. *Genome Res. v. 19*. p. 1639-1645.

Li, Na. (2005). Applications for Running DOS Commands within SAS®. *www.lexjansen.com/pharmasug/2005/posters/po13.pdf*.

Murphy, William C., Proskin, Howard M. & Associates Inc. (2008). Tools of Miss-Calculation: Managing Missing Values with SAS®. *SAS® Global Forum 2008.*

RESOURCES

http://strawberry-perl.googlecode.com/files/strawberry-perl-5.10.1.3.msi

http://mkweb.bcgsc.ca/circos/distribution/circos-**0.52.t**gz

http://mkweb.bcgsc.ca/circos/distribution/circos-**0.52-2.**tgz

http://www.census.gov/population/www/socdemo/state-to-state.html

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Danni Bayn Ph.D.
Capella University
225 S 6th St, 9th Fl
Minneapolis, MN 55402
Danni.Bayn@capella.edu
http://www.capella.edu

‘CIRCOS TO SAS®’ SOURCE CODE

```
/********************************************************************************\
PROGRAM INFORMATION
Project             : Circos & SAS
Purpose             : To install and run Circos Tableviewer
Requirements        : Have Perl installed and have the Circos files extracted to a folder within the Perl directory
Outputs             : Configuration files, Perl configuration file, Final Table

PROGRAM HISTORY
2010-10-15 Danni Bayn Initial program developed.
\**********************************************************************************/;

%let CIRCOS_PATH = C:\strawberry\perl\circos-0.52\tools\tableviewer;
%let RAW = H:\reporting\ad_hoc\Internal Transfer Summary.txt;
%let DATA_FILE = H:\reporting\ad_hoc\Circos.txt;
%let OUT = H:\reporting\ad_hoc\Graph.svg;

%Install_Circos()
%Sample_Graph()
%Customize_graph(Yes, FONT_SIZE=24, RADIUS=0.60, RIBBON_VAR=no)
%Create_Graph(work.states)

/********************************************************************************\
Macro to install required Perl modules and to Build Circos.
\********************************************************************************/;
%macro Install_Circos ();
/*  Install Dependencies*/
  x "ppm install MinGW";
  x "ppm install clone";
  x "ppm install Config::General";
  x "ppm install Math::Bezier";
  x "ppm install Math::Round";
  x "ppm install Math::VecStat";
  x "ppm install Params::Validate";
  x "ppm install Set::IntSpan";
  x "ppm install Statistics::Descriptive";
  x "ppm install Graphics::ColorObject";
  x "ppm install Class::Base";
  x "ppm install Math::Random";
  x "ppm install Readonly";

/*  Build/Install Circos*/
  x "cd C:\Strawberry\Perl\circos-0.52";
  x "perl build.pl";
  x "build manifest";
  x "build.bat";
  x "cd &CIRCOS_PATH";
  x "cd etc\";

/*  Create copies of original conf files*/
  x "copy circos.conf circos.conf.bak ";
  x "copy make-table.conf make-table.conf.bak ";
  x "copy parse-table.conf parse-table.conf.bak ";
  x "copy ideogram.conf ideogram.conf.bak ";
  x "cd ..";
%mend Install_Circos;
```

```
/**********************************************************************************\
  Macro to create a sample data set and graph.  Make sure to have defined
    &CIRCOS_Path and &OUT before running.
\**********************************************************************************/;
%macro Sample_Graph ();
  x "perl bin/make-table -rows 10 -unique -brief | perl bin/parse-table | perl bin/make-conf -dir data";
  data _null_;
    wait_sec=sleep(20);
  run;
  x "perl ../../bin/circos -conf etc/circos.conf";
  data _null_;
    wait_sec=sleep(20);
  run;
  x "%str(copy %"&CIRCOS_PATH\tableview.svg%" %"&OUT%" )";
%mend Sample_Graph;

/**********************************************************************************\
  Macro to take a custom dataset and generate a graph. Make sure to have defined
    &CIRCOS_PATH
    &DATA_FILE
    &OUT
\**********************************************************************************/;
%macro Create_Graph (DATASET);
  filename out "&DATA_FILE" lrecl=5000;
  proc export data =&DATASET replace
    outfile=out
    dbms=dlm;
    delimiter='09'x;
  run;
  filename out clear;

  x "cd &CIRCOS_PATH";
  x "type &DATA_FILE | perl bin/parse-table | perl bin/make-conf -dir data";

  data _null_;
    wait_sec=sleep(20);
  run;

  x "perl ../../bin/circos -conf etc/circos.conf";
  data _null_;
    wait_sec=sleep(20);
  run;
  x "%str(copy %"&CIRCOS_PATH\tableview.svg%" %"&OUT%" )";
%mend Create_graph;

/**********************************************************************************\
  Macro to customize the configuration files.
\**********************************************************************************/;
%macro Customize_graph (
  MOD_ORIG
  , FONT_SIZE
  , RADIUS
  , RIBBON_VAR
  );

  %if %upcase("&MOD_ORIG") eq "HELP" %then %do;
    %put **************************************************************************************************;
    %put * CUSTOMIZE_CIRCOS Modifies the conf files used in creating circular graphs                   ;
    %put **************************************************************************************************;
    %put * Positional Parameters (in this order):                                                      ;
    %put *  MOD_ORIG    Defaults to YES. Determines whether you want to create new copies of the Conf   ;
    %put *          files off of the original backups (YES) or build off of modifications that have ;
    %put *          already been made (NO).                                                         ;
    %put *                                                                           ;
    %put * Optional Keyword Parameters (in any order):                                              ;
    %put *  FONT_SIZE    Enter a numeric value to set the size of the Labels (default is 48).            ;
    %put *  RADIUS       Determines the radius of the graph compared to the size of the image. Default  ;
    %put *               is 0.85 (percentile). Lower this if your labels are getting cut off            ;
    %put *  RIBBON_VAR   Changes whether or not the ends of the ribbons are variable. Default is NO     ;
    %put *********************************************************************************************;
    %put * Example macro call (remove space after %)                                          ;
    %put * % customize_graph(NO, FONT_SIZE=36, RADIUS= 0.79, RIBBON_VAR = yes );
    %put *********************************************************************************************;
    dm log 'show';
    %goto ByeBye;
  %end;

  %if %upcase("&MOD_ORIG") eq 'NO' %then %do;
    %let CIRCOS = C:\strawberry\perl\circos-0.52\tools\tableviewer\etc\Circos.conf;
    %let IDEO = C:\strawberry\perl\circos-0.52\tools\tableviewer\etc\ideogram.conf;
    %let PARSE = C:\strawberry\perl\circos-0.52\tools\tableviewer\etc\parse-table.conf;
  %end;
  %else %do;
    %let CIRCOS = C:\strawberry\perl\circos-0.52\tools\tableviewer\etc\circos.conf.bak;
    %let IDEO = C:\strawberry\perl\circos-0.52\tools\tableviewer\etc\ideogram.conf.bak;
    %let PARSE = C:\strawberry\perl\circos-0.52\tools\tableviewer\etc\parse-table.conf.bak;
  %end;

/*  Creates perl configuration file to open and modify other Conf files.*/
   x "cd &CIRCOS_PATH";
  data _null_;
   file 'etc\Configuration.pl';
    put '#!/usr/bin/perl;';
    put 'use strict;';
    put "my $font_size = '&FONT_SIZE.p';";
    put "my $radius = '&RADIUS.r';";
    put "my $ribbon = '&RIBBON_VAR';";
    put "open(CIRCOS, '&CIRCOS'); ";
    put "open(IDEO, '&IDEO'); ";
    put "open(PARSE, '&PARSE'); ";
    put "open(OUT_CIRCOS, '>circos.conf'); ";
    put "open(OUT_IDEO, '>ideogram.conf'); ";
    put "open(OUT_PARSE, '>parse-table.conf'); ";
    put '#Font_Size';
    put 'foreach(<CIRCOS>) {';
    put ' if($font_size) {';
    put '   if(/^label_size\s=/) {';
    put '     print OUT_CIRCOS "label_size = $font_size\n";';
    put '   } else { print OUT_CIRCOS "$_"; } ';
    put ' } else { print OUT_CIRCOS "$_"; } }';
    put '#Radius';
    put 'foreach(<IDEO>) {';
    put ' if($radius) {';
    put '   if(/^radius\s+=/) {';
    put '     print OUT_IDEO "radius\t = $radius\n";';
    put '   } else { print OUT_IDEO "$_"; }';
    put ' } else { print OUT_IDEO "$_"; } }';
    put '#Ribbon';
    put 'foreach(<PARSE>) {';
    put ' if($ribbon) {';
    put '   if(/^ribbon_variable = /) {';
    put '     print OUT_PARSE "ribbon_variable = $ribbon\n";';
    put '   } else { print OUT_PARSE "$_";}';
    put ' } else { print OUT_PARSE "$_"; } }';
    run;
    x "cd etc";
    x "perl configuration.pl";
%ByeBye:;
%mend Customize_graph;


/**********************************************************************************\
   Sample code to prepare data. Presumes initial dataset is formatted as
            To                  From                Count
            Alabama             Alaska              2912
            Alabama             Arizona             896
            ….

            Reformats the data to the format required for Circos.
                                Alabama             Alaska
            Alabama             -                   2912
            Alaska              1392                -
            …
\**********************************************************************************/;

filename nca "&RAW" lrecl=5000;

data rwork.Circos;
  infile nca dlm='09'x dsd missover firstobs=1;
  attrib
    Program1 format=$75. label='To'
    Program2 format=$75. label='From'
    Count format=7. label='Count'
;
  input Program1--Count;
run;
filename nca clear;

option missing ='-';
rsubmit;
data circos_parse;
  set circos;
  Program1 = prxchange('s/\s+//', -1, Program1);
  Program2 = prxchange('s/\s+//', -1, Program2);
run;

proc sort data=circos_parse;
  by Program1 Program2;
run;

proc transpose data=circos_parse out=circos_t (drop=_name_ _label_);
  by Program1;
  ID Program2;
  Var Count;
run;
endrsubmit;
```