ITAR User Guide

Analyzing Information Theory Results Generated Using TIPNet

Introduction

This set of programs allows for the analysis of hydroelectric reservoir data via information theory (IT), and for representation of the results in alternative formats. IT analysis is performed using a Temporal Information Partitioning Networks (TipNET) program developed by Allison Goodwell and Praveen Kumar. The python programs described here can then be used to represent the data in two different ways. The excel file produced by *Creating IT Tables from TIPNets Results.py* contains tables which provide detailed information about the amount and type of information provided by each particular link in an easy-to-interpret format. The images produced using Circos (with files generated by *Creating Circos Files from IT Tables.py*) show the overall information contribution of reservoirs and pairs of reservoirs to a particular target variable. These two representations together provide a more complete picture of the information theory results produced with TIPNet, increasing their utility.

TIPNet (Matlab)

Install TIPNet: https://github.com/HydroComplexity/TIPNet

Run EntropyGUI_mainwindow.m in Matlab

Upload excel file with data for analysis; keep default settings; click compute links

The excel file should have an extra column (with any data) at the end so all columns load properly

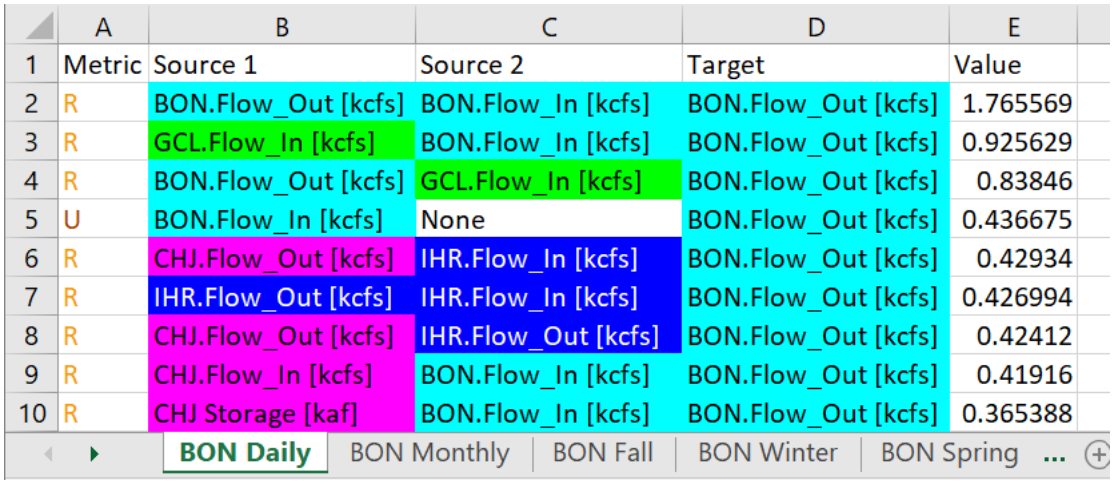
Use the Columba_Timestep.xlsx files to generate the same output as is in the example files

Export TIPNet Results to CSV Files

- Export: pairvars, varnames, Itot_allpairs, R_allpairs, S_allpairs, and U (or whichever IT metrics you want to use) for each timestep
    - pairvars contains the first source (column one) and second source (column two) informing each target
    - varnames contains the variable names in the order in which they're numbered for use in pairvars
- writematrix(sprsheet name, 'FileName.csv')
    - writecell(sprsheet name, 'FileName.csv') for varnames, since it's a single line
- NOTE: Use the same naming convention for each filename (aside from varnames and pairvars)
- Save files in same folder as python programs

Creating IT Tables from TIPNets Results.py

This program takes the information theory results from TIPNet, assembles the source pairs with their targets, groups the targets by reservoir, and assembles them into color-coded rows. The rows are ordered such that the sourcepair-target combination which produces the highest-valued information theory metric is at the top of the table, followed by the next highest, and so on. An example output is shown below, in *Figure 1*.



	A	B	C	D	E
1	Metric	Source 1	Source 2	Target	Value
2	R	BON.Flow_Out [kcfs]	BON.Flow_In [kcfs]	BON.Flow_Out [kcfs]	1.765569
3	R	GCL.Flow_In [kcfs]	BON.Flow_In [kcfs]	BON.Flow_Out [kcfs]	0.925629
4	R	BON.Flow_Out [kcfs]	GCL.Flow_In [kcfs]	BON.Flow_Out [kcfs]	0.83846
5	U	BON.Flow_In [kcfs]	None	BON.Flow_Out [kcfs]	0.436675
6	R	CHJ.Flow_Out [kcfs]	IHR.Flow_In [kcfs]	BON.Flow_Out [kcfs]	0.42934
7	R	IHR.Flow_Out [kcfs]	IHR.Flow_In [kcfs]	BON.Flow_Out [kcfs]	0.426994
8	R	CHJ.Flow_Out [kcfs]	IHR.Flow_Out [kcfs]	BON.Flow_Out [kcfs]	0.42412
9	R	CHJ.Flow_In [kcfs]	BON.Flow_In [kcfs]	BON.Flow_Out [kcfs]	0.41916
10	R	CHJ Storage [kaf]	BON.Flow_In [kcfs]	BON.Flow_Out [kcfs]	0.365388

BON Daily | BON Monthly | BON Fall | BON Winter | BON Spring ...

Figure 1. Example IT Table Output. Tables are grouped by reservoir and timestep, and all rows have a single target variable.

Features

The information theory metrics and timesteps displayed in the table can be defined in IT_metric_lst and and timestep_lst, respectively. All reservoirs used to generate TIPNet results must be included in allmatlabused_reservoir_lst. The reservoirs which should be included as targets in the final spreadsheet can be defined using alloutput_reservoir_lst.

WHAT TO CHANGE BEFORE RUNNING
(Generally preceded by the comment `***CHANGE TO FIT DATA***`):

- Change allmatlabused_reservoir_lst to match the reservoirs used in TIPNet analysis, in the order they appear in varnames
- Change target_string if desired. Ensure the target is of the form Reservoir + target_string
    - Ex. Reservoir = BON, target_string = '.Flow_Out [kcfs]' → Target in spreadsheet = 'BON.Flow_Out [kcfs]'
- Change sprsheet_name to change the name of the output excel file
- Delete/change PDO_yes_no to fit data

- Change file_str following the comment `# Creating lists of filenames for each IT metric and timestep` to match the TIPNet data filenames
    - The filenames should contain the timestep and IT metric of the results

NOTE: Create a version of the excel file with all reservoirs used to generate the IT results. Use this as the input for Creating Circos Files from IT Tables.py

Creating Circos Files from IT Tables.py

This program reads in the data from the excel file produced by Creating IT Tables from TIPNets Result.py and outputs Circos karyotype and link files (grouped by reservoir and timestep) which can used to generate circular plots of the information theory results. Information is summed such that information sourcepairs in the Circos files are from two particular reservoirs (regardless of variable type). This information is then converted into the required format for Circos. The link strengths are multiplied by a factor of 10 and converted to integers to meet the requirements for Circos link sizing. An example of this process is shown in *Figure 2*.

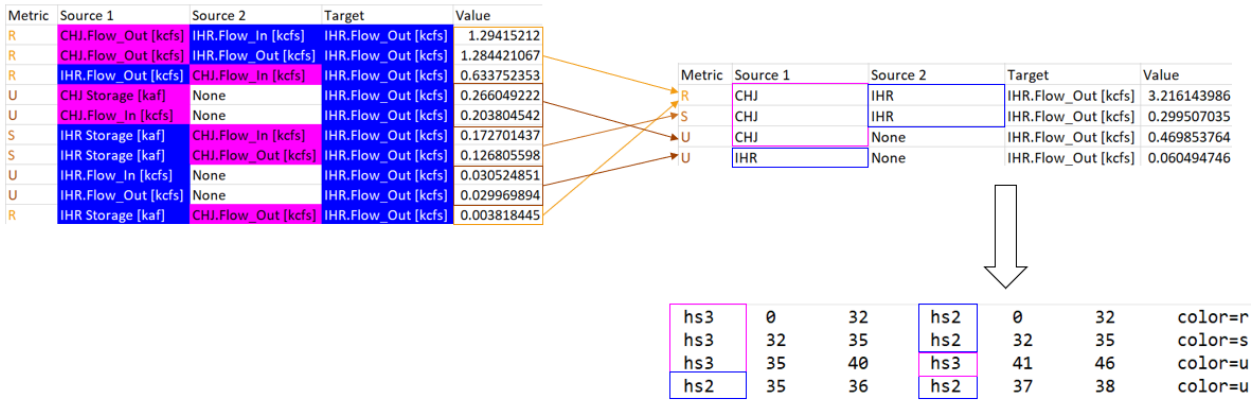


Figure 2. General process by which the IT table data is converted to Circos files by Creating Circos Files from IT Tables.py

WHAT TO CHANGE BEFORE RUNNING:

- Make sure reservoir_lst and timestep_lst match alloutput_reservoir_lst and timestep_lst in Creating IT Tables from TIPNets Results.py
- Use an unchanged excel file outputted by Creating IT Tables from TIPNets Results.py as the input_file
- Make sure this program can access the same varnames.csv file used in Creating IT Tables from TIPNets Results.py; change the name in this program if necessary
- Change the link and karyotype file names (called link_file and karyotype_file) near the end of the program if desired

Running Circos

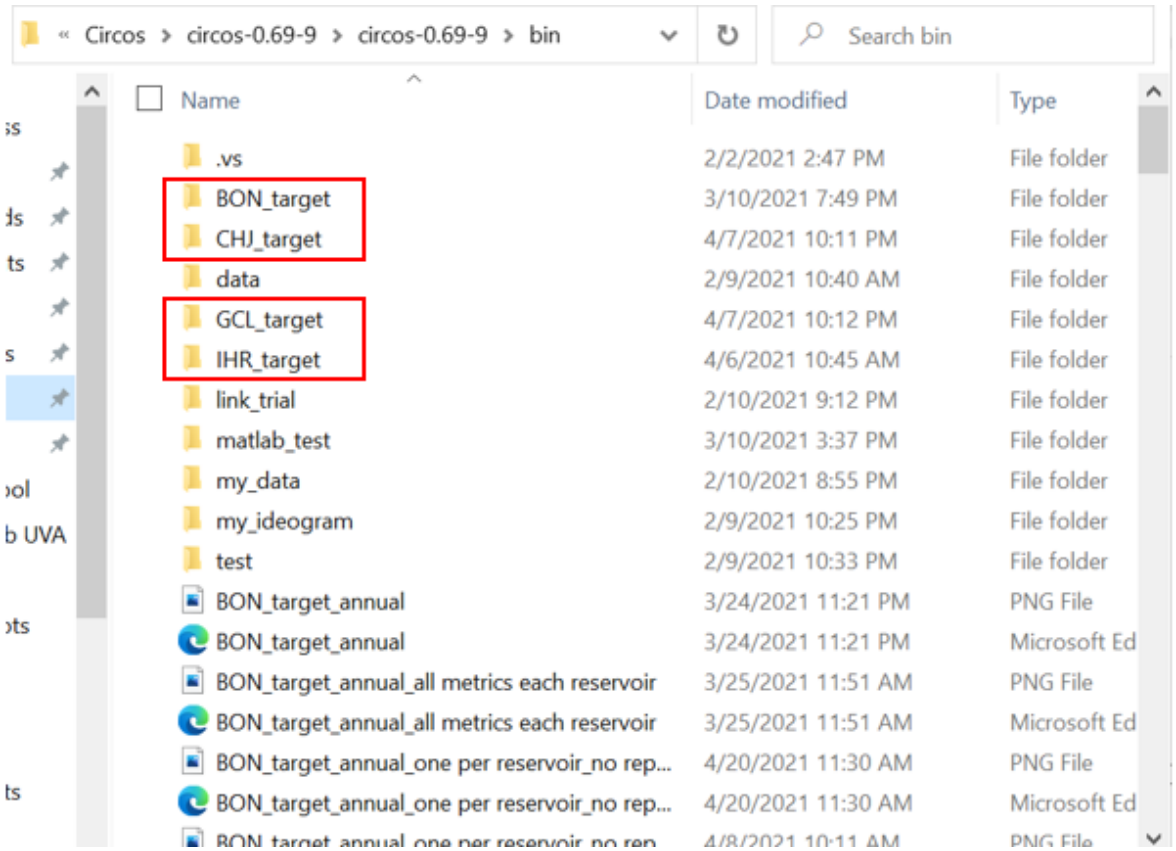
Install and test Circos

http://circos.ca/software/download/

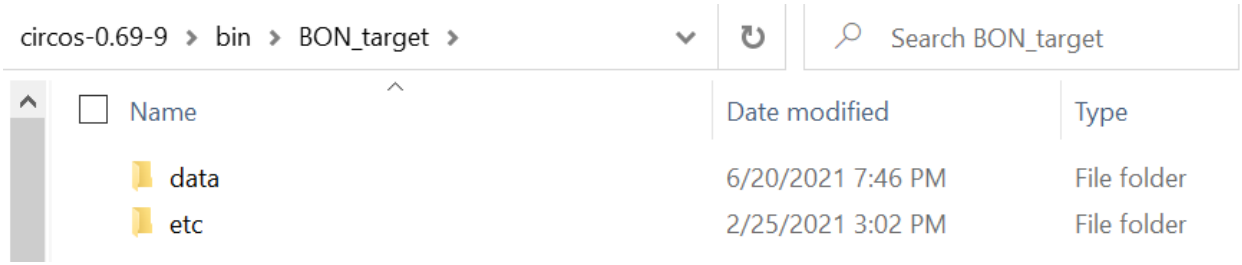
https://www.youtube.com/watch?v=8fCEhUwkNm0

https://www.youtube.com/watch?v=4n_Ckdhirnw

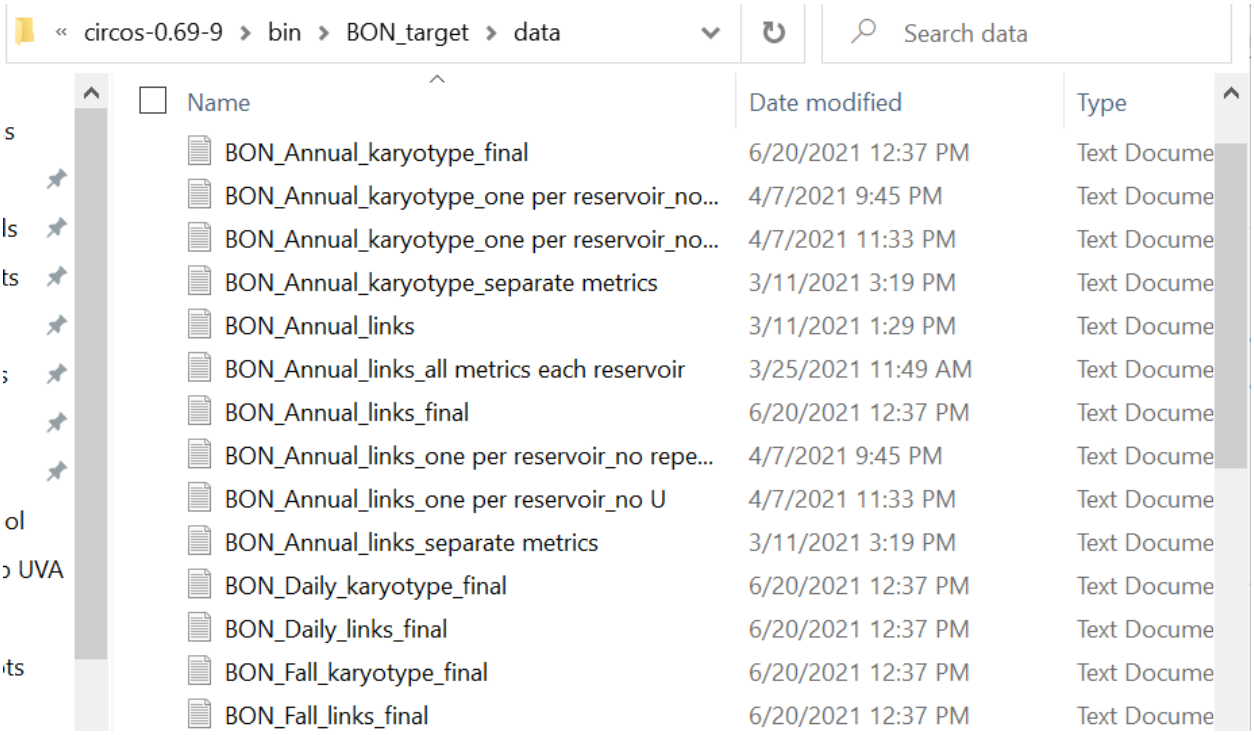
Within the bin folder, create files for each reservoir (data from this reservoir will be the target in these files)



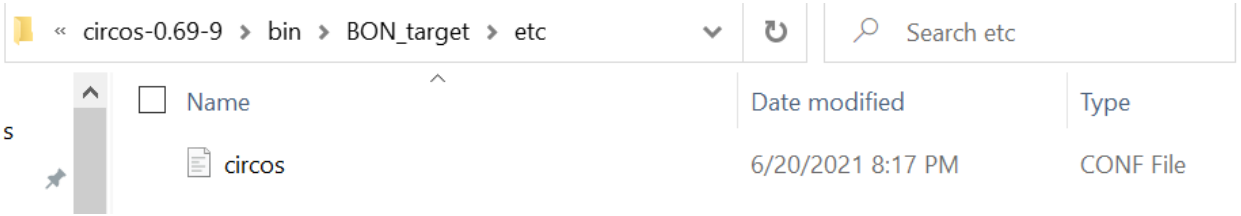
Within the folder for each reservoir, create a data and etc folder



Data will contain the karyotype and link files for the reservoir (for each timestep) outputted by Creating Circos Files from IT Tables.py

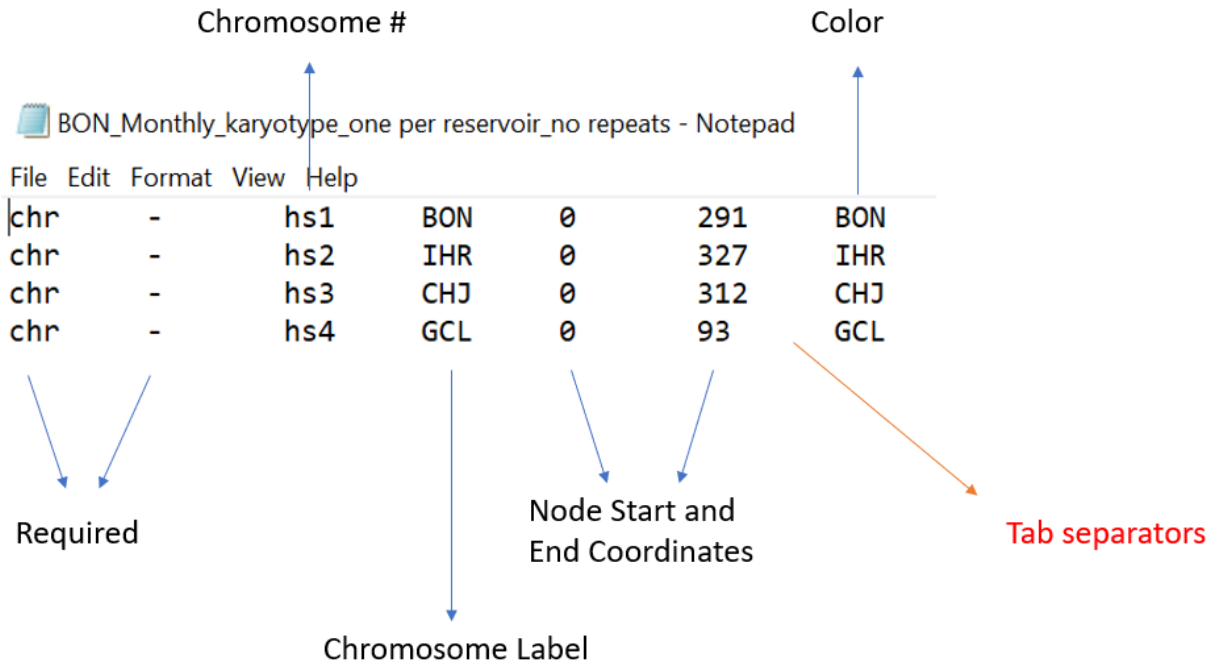


Etc will contain the configuration file for the reservoir. This file will need to be modified for each timestep when generating images.



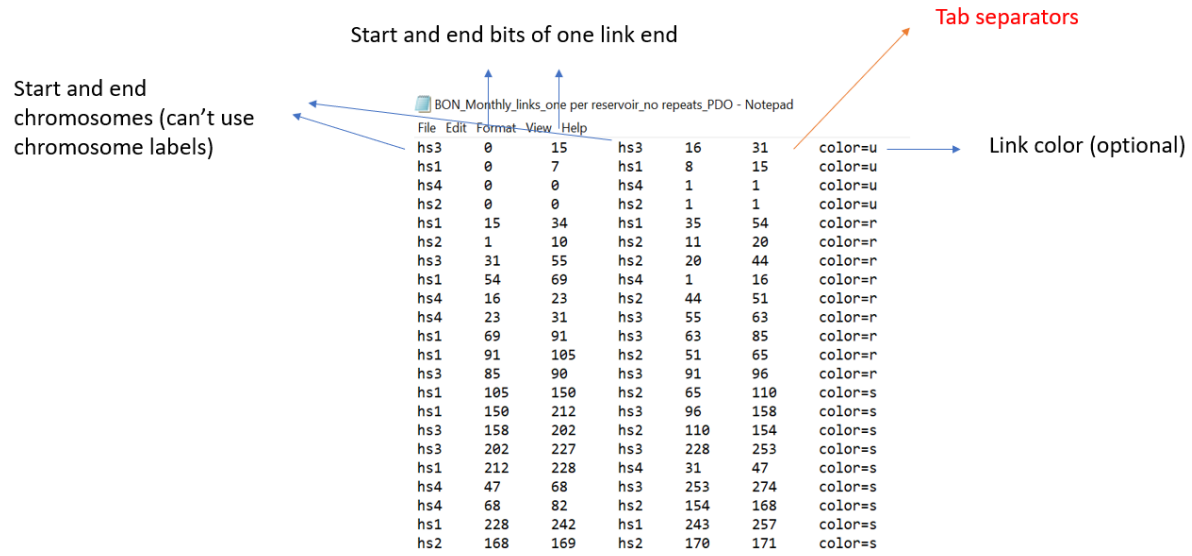
Karyotype File

A text file which defines the size, color, and label of each node. Nodes are referenced using their "chromosome number," defined using the letters hs followed by the number.



Link File

Defines the width and color of links between reservoir (chromosomes in Circos). The width corresponds with the strength of the link, while the color indicates the information type (U, R, or S).

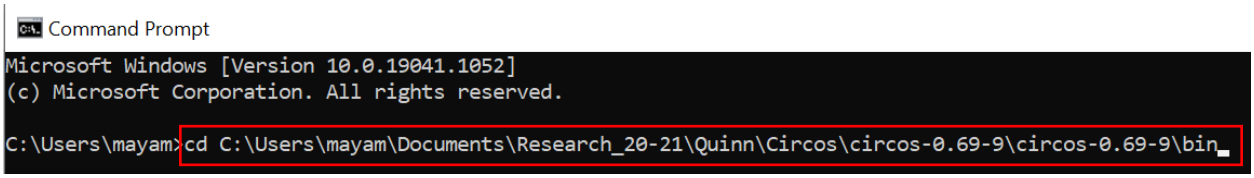


Configuration File (circos.conf)

This is the file which Circos accesses to generate the image. Can use this file as is (make sure it is in the etc folder for the reservoir(s) of interest), just change the filename and timestep as needed before running Circos.

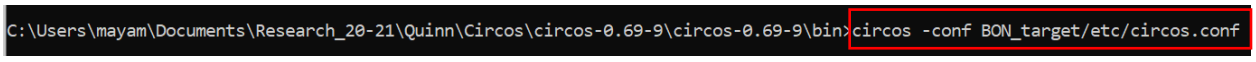
Generating Images

Within the terminal, type cd followed by the file path to bin

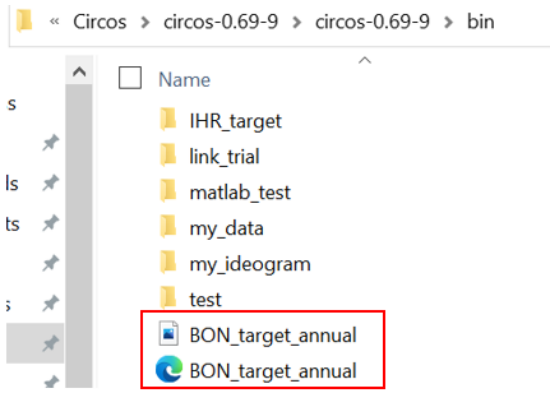


Then, run Circos

Command: circos -conf FOLDER/etc/circos.conf



Access the final image within the bin folder



Troubleshooting

Python Programs

Check/change python's current working directory using os if there are errors with Creating IT Tables from TIPNets Results.py or Creating Circos Files from IT Tables.py

Ensure the xlrd package is installed and installed properly

Circos

Make sure the configuration file has the correct file names and is saved between runs. Use the Circos Google group or documentation for other problems