Match Structures Changes as the FIRST Robotics Competition Program Evolved

Network Analysis of FRC Matches

Jonas Wagner

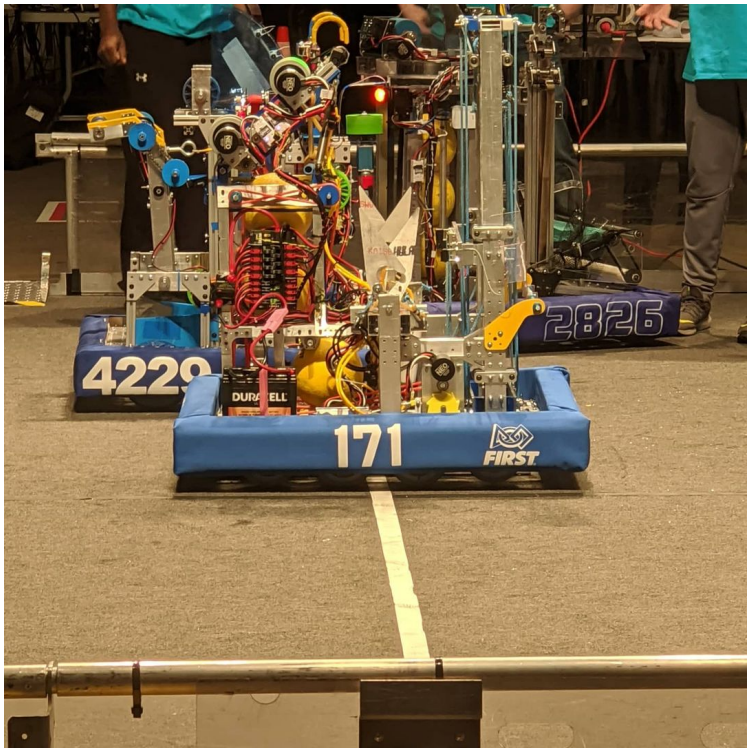
Univeristy of Texas at Dallas

SYSM 6302 Final Project
2021 May 7th

Table of Contents

FIRST Robotics Competition

FRC 2826 - Wave Robotics - Oshkosh, WI
World Championship Finalists 2015



FRC 171 - The Cheese Curd Herd - Platteville, WI
Northern Lights Regional 2020 (Duluth, MI)

Table of Contents

The Blue Alliance

www.thebluealliance.com [1]

The Blue Alliance is a very useful online database of past (and current) match data.

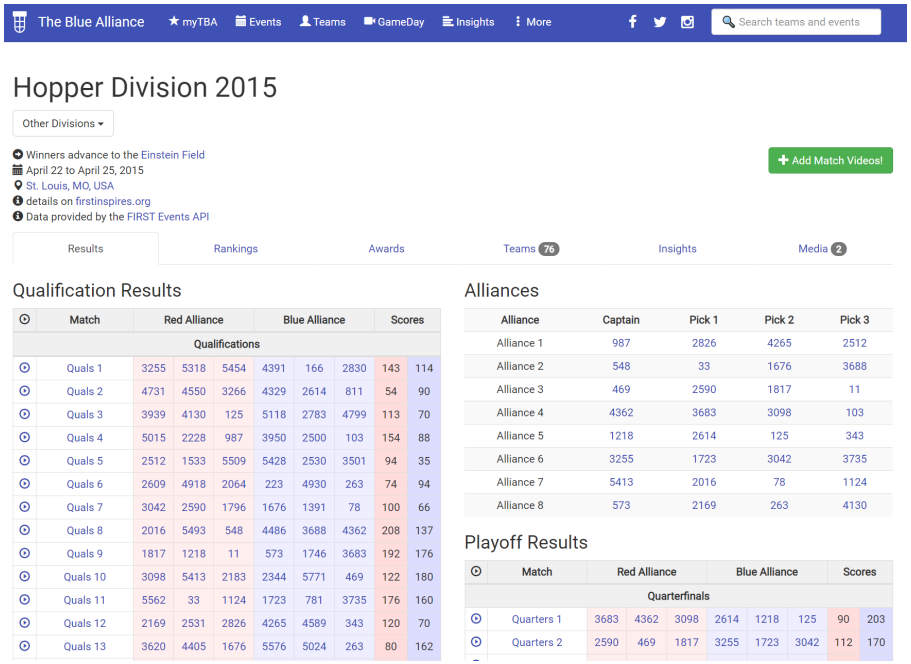


Table of Contents

Accessing The Blue Alliance Database

...nas\OneDrive - The University of Texas at Dallas\2021_Spring\SYSM6302\Project\TBA_database_access.py

test.py | TBA_database_access.py | TBA_Network_Analysis.py | dist_and_centrality_calc.py | graph_visualize.py

```python
# -*- coding: utf-8 -*-
"""
TBA_database_access is a module with functions for requesting from TBA.

Created on Wed Feb 24 18:43:49 2021

@author: Jonas
"""
import requests

# Interaction with TBA database
def getTBA(url, baseURL = 'http://www.thebluealliance.com/api/v3/',
           header = {'X-TBA-Auth-Key': '7uiNdsPDiLnsKapBtr3IlZfb4wYGkvpPD9IpyC
    """
    Get the JSON from The Blue Alliance database for requested url extension.

    Parameters
    ----------
    url : string
        url request extension.
    baseURL : string, optional. Default is TBA API v3
    header: set of strings, optional. Default is just TBA auth key.

    Returns
    -------
    json
        Requested data from The Blue Alliance.

    """
    return requests.get(baseURL + url, headers = header).json()


# Event Specific Queries
def getEventKeys(year):
    """
    Get all the keys of events over the course of a year

    Parameters
    ----------
    year : str (or int)

    Returns
    -------
    list of str (match_keys)

    """
    return getTBA('events/' + str(year) + '/keys')
```

```python
# Match Specific Queries
def getMatchData(key, simple = True):
    """
    Get the match data for a particular match

    Parameters
    ----------
    key : str (match_key)
    simple : bool, optional
        simple match data or not. The default is True.

    Returns
    -------
    matchData : dict
        dict containing all the match data

    """
    if simple == True:
        matchData = getTBA('match/' + key + '/simple')
    else:
        matchData = getTBA('match/' + key)
    return matchData
```

Table of Contents

Network Generation

...as\OneDrive - The University of Texas at Dallas\2021_Spring\SYSM6302\Project\TBA_Network_Analysis.py

test.py | TBA_database_access.py | TBA_Network_Analysis.py* | dist_and_centrality_calc.py | graph_visualize.py

```
10  # Nessicary Packages
11  import networkx as nx
12  import itertools
13  import TBA_database_access as tba
14  import numpy as np
15  from os import path
16  import matplotlib.pyplot as plt
17  import time
18
19  ## TBA_Network class ------------------------------------------------------
20  class TBA_Network:
21      """
22      TBA_NETWORK is a class for creating and analyzing networks made up from
23      FIRST Robotics Competition match data from The Blue Alliance.
24
25      Atributes
26      ---------
27      year : str
28          year of frc season.
29      event : list of str
30          list of event_keys for all events in network.
31      nodeType : str
32          node type (only programmed for teams as nodes).
33      edgeType : str
34          edge type (only programmed for matches as edges).
35      nodeMeta : list of str
36          list of meta elements to be stored as node attibutes.
37      edgeMeta : list of str
38          list of meta elements to be stored as edge attibutes.
39      nodeKeys : list of str
40          list of keys for all the nodes in the network.
41      edgeKeys : list of str
42          list of keys for all the (meta-)edges in the network.
43          (These edges are actually more like clusters... each is a match)
44      nodeData : dict of dicts
45          dictionary of all node attributes keyed by the nodeKeys.
46      edgeData : dict of dicts
47          dictionary of all edge attributes keyed by the nodeKeys.
48          (These edges are actually more like clusters... each is a match)
49      edgeTuples : list of tuples
50          list containing tuples related to every node within a the network.
51          (used to generate all the edges within the nx.MultiGraph)
52      G : nx.MultiGraph
53          nx.MultiGraph constructed with all the nodes and edges with the metadata
54          stored as attributes.
55      G_default : nx.Graph
56          nx.Graph that is the default weighted undirected projection representing
57          the total number of edges that connects them.
58      G_----- : nx.Graphs
59          many additional projections will be saved to the network class itself
60
61      Methods
62      -------
```

```
    Methods
    -------
    GetProjection(self, projection = 'none')
        Locates or generates a specific projection
    GraphProjections(self, alliancePartners = 0, weightCalc = 'default')
        Generates undirected weighted projections of the network with weighting
        calculated acording to specific parameters.
    WeightCalc(self, team1, team2, weightCalc)
        Function for detirmining the weighting between two nodes.
    TeamKeys(self)
        Method that returns a list of team keys
    MatchKeys(self)
        Method that returns a list of match keys

    Centrality(self, node=-1, projection='none', mode='degree', print_top_nodes=-1)
        Calculates the a particular centrality metric fo a specified projection
    Diameter(self, projection='none')
        Calculates the diameter of a particular projection
    DegreeSequence(self, projection = 'none')
        Returns a sorted list with team keys and degrees
    DegreeDist(self, projection='none')
        Returns a list of the degree distribution
    PlotDDist(self, projection='none', axes=-1)
        Plots a Degree Distribution for a particular projection
    DrawGraph(self, projection='none', -------)
        Draw a simple projection of the graph using built in commands

    """

    def __init__(self, year = -1, event = -1,
                 nodeType = 'team', edgeType = 'match',
                 nodeMeta = ['nickname', 'name', 'state_prov', 'country'],
                 edgeMeta = ['match_key', 'event_key', 'scores', 'teams',
                             'winning_alliance', 'alliancePartners',
                             'comp_level', 'match_number'],
                 filename_base = 'Data/TBA_Network_',
                 ):
        """
        TBA_NETWORK constructor function. Creates a TBA_Network object
        constructed with TBA data acorrding to specific inputs

        Parameters
        ----------
        year : int or str, optional
            year specifier. The default is -1.
```

Table of Contents

Full Season Network

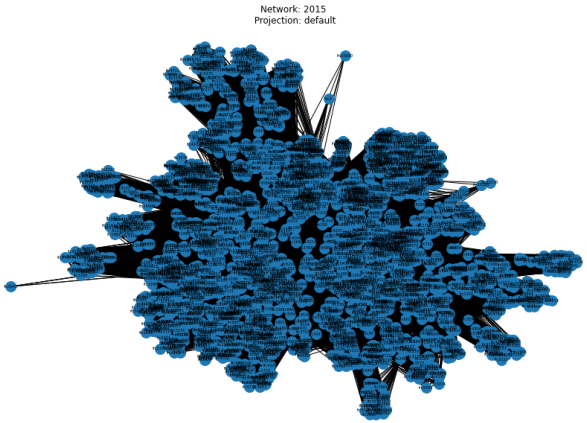
2015 (my favorite season)

Network: Entire 2015 Season

TBA_Network_2015.gml (45 MB)

(https://www.thebluealliance.com/events/2015)

- Nodes: 2,935 Teams
- Mult-edges: 195,152 Matches
- Edges: 106,889 Unique Matches
- Components: 1
- Diameter: 4
- Clustering Coefficient: 0.635
- Degree Assortativity: 0.296 (0.358)
- Average Degree: 73.49 (132.98)
- Density: 0.02527 (0.04532)



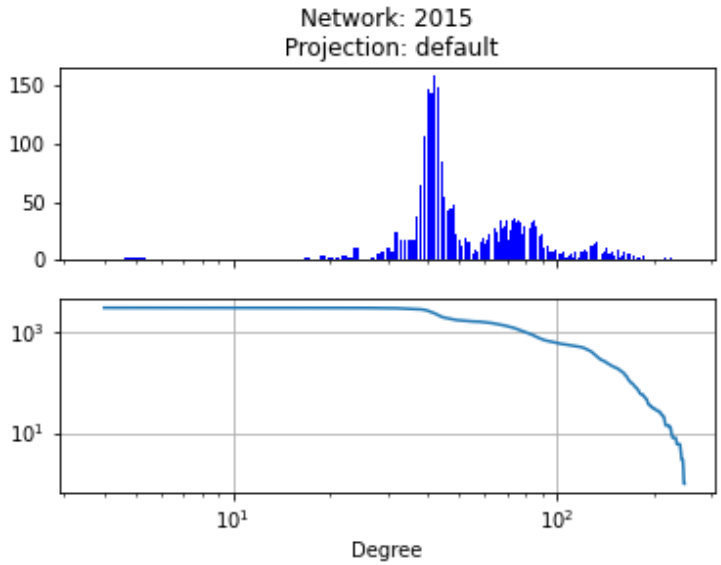


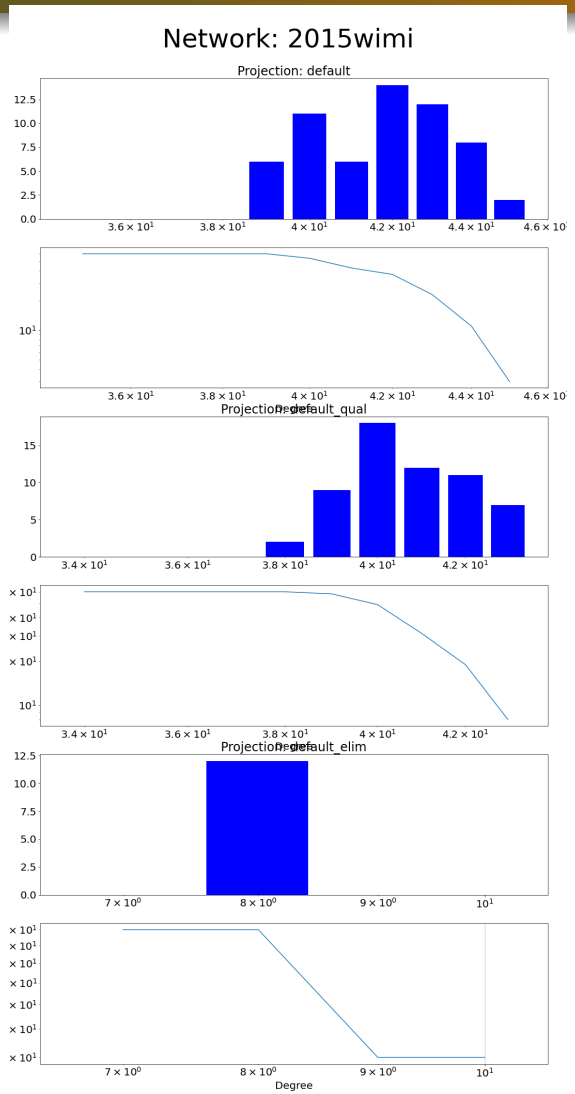
Table of Contents

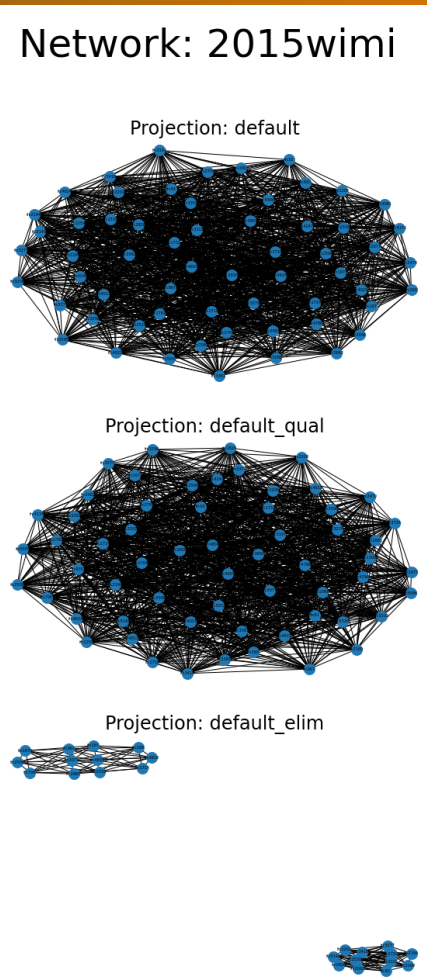
Table of Contents

2015 Wisconsin Regional Analysis

Milwaukee, WI

```
Getting Projection: default
Diameter =  2
----------------------------------------------
Getting Projection: default
degree centrality
----------------------
  1. frc706 (0.7797)
  2. frc269 (0.7627)
  3. frc3102 (0.7627)
  4. frc1732 (0.7458)
  5. frc2077 (0.7458)
  6. frc2506 (0.7458)
  7. frc2883 (0.7458)
  8. frc4818 (0.7458)
  9. frc967 (0.7458)
  10. frc167 (0.7458)
----------------------
Getting Projection: default
eigenvector centrality
----------------------
  1. frc706 (0.1419)
  2. frc3102 (0.1389)
  3. frc269 (0.1389)
  4. frc2506 (0.1361)
  5. frc2077 (0.1358)
  6. frc1732 (0.1356)
  7. frc2883 (0.1355)
  8. frc967 (0.1355)
  9. frc167 (0.1354)
  10. frc4818 (0.1352)
----------------------
Getting Projection: default
katz centrality
----------------------
  1. frc706 (0.1864)
  2. frc3102 (0.1806)
  3. frc269 (0.1755)
  4. frc2506 (0.1650)
  5. frc2826 (0.1641)
  6. frc2077 (0.1629)
  7. frc1732 (0.1627)
  8. frc2883 (0.1602)
  9. frc1675 (0.1595)
  10. frc93 (0.1571)
----------------------
```

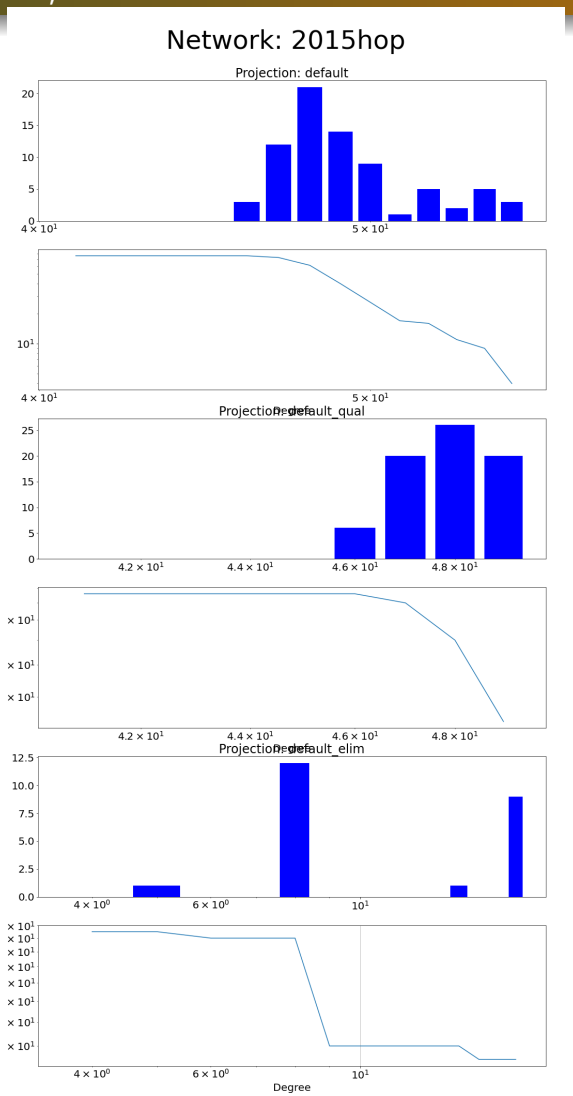


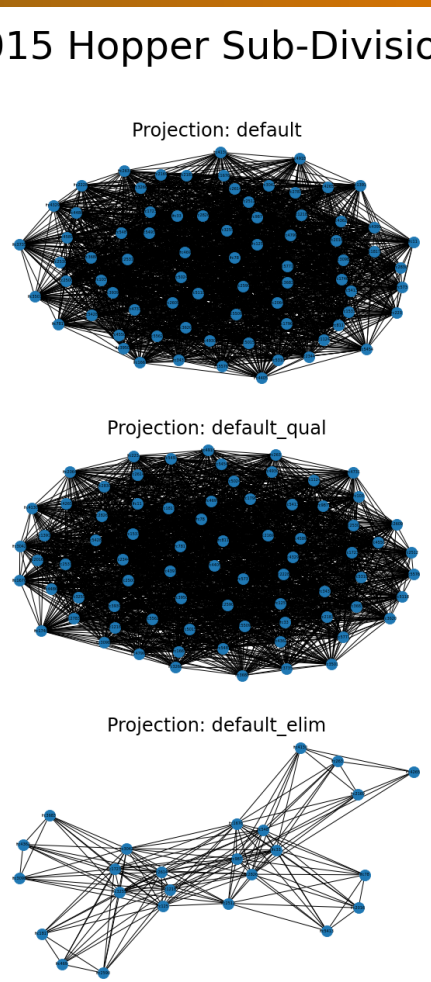


2015 Hopper Sub-Division Analysis

FIRST Championships 2015: St. Louis, MO

```
Getting Projection: default
Diameter =  2
--------------------------------------------
Getting Projection: default
degree centrality
----------------------
  1. frc3042 (0.7467)
  2. frc1676 (0.7333)
  3. frc1723 (0.7333)
  4. frc125 (0.7333)
  5. frc3255 (0.7200)
  6. frc548 (0.7200)
  7. frc987 (0.7200)
  8. frc2512 (0.7200)
  9. frc33 (0.7200)
  10. frc2826 (0.7067)
----------------------
Getting Projection: default
eigenvector centrality
----------------------
  1. frc3042 (0.1303)
  2. frc1676 (0.1284)
  3. frc1723 (0.1281)
  4. frc125 (0.1280)
  5. frc3255 (0.1263)
  6. frc987 (0.1262)
  7. frc2512 (0.1260)
  8. frc33 (0.1260)
  9. frc548 (0.1257)
  10. frc2826 (0.1237)
----------------------
Getting Projection: default
katz centrality
----------------------
  1. frc987 (0.2103)
  2. frc3255 (0.2103)
  3. frc1676 (0.2078)
  4. frc3042 (0.1954)
  5. frc33 (0.1936)
  6. frc1723 (0.1912)
  7. frc548 (0.1883)
  8. frc2512 (0.1864)
  9. frc2826 (0.1839)
  10. frc2614 (0.1804)
----------------------
```

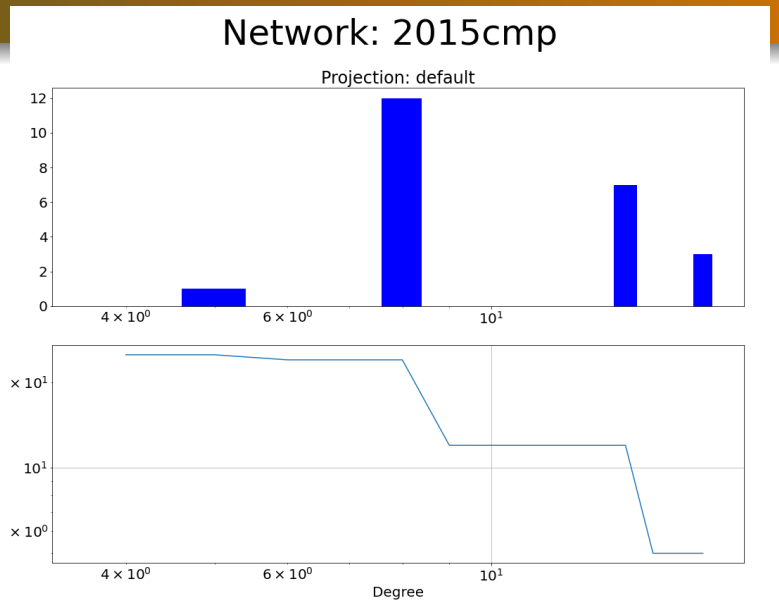


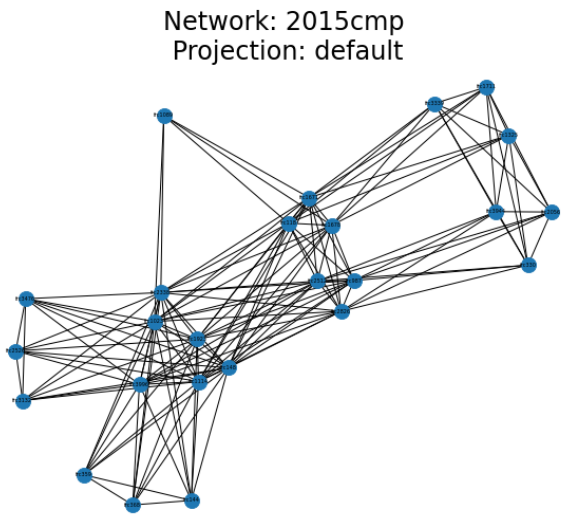


2015 Einstein Analysis

FIRST Championships 2015: St. Louis, MO

```
Getting Projection: default
Diameter =  3
-----------------------------------------------
Getting Projection: default
degree centrality
----------------------
  1. frc1023 (0.7500)
  2. frc2338 (0.7500)
  3. frc1114 (0.7083)
  4. frc148 (0.7083)
  5. frc1923 (0.7083)
  6. frc118 (0.5833)
  7. frc1671 (0.5833)
  8. frc1678 (0.5833)
  9. frc2512 (0.5833)
  10. frc2826 (0.5833)
----------------------
Getting Projection: default
eigenvector centrality
----------------------
  1. frc1023 (0.2931)
  2. frc2338 (0.2931)
  3. frc1114 (0.2861)
  4. frc148 (0.2861)
  5. frc1923 (0.2861)
  6. frc2512 (0.2397)
  7. frc2826 (0.2397)
  8. frc987 (0.2397)
  9. frc3996 (0.2370)
  10. frc118 (0.2293)
----------------------
Getting Projection: default
katz centrality
----------------------
  1. frc1023 (0.3181)
  2. frc2338 (0.3181)
  3. frc1114 (0.3122)
  4. frc148 (0.3122)
  5. frc1923 (0.3122)
  6. frc3996 (0.2535)
  7. frc2826 (0.2338)
  8. frc987 (0.2338)
  9. frc2512 (0.2338)
  10. frc1678 (0.2152)
----------------------
```





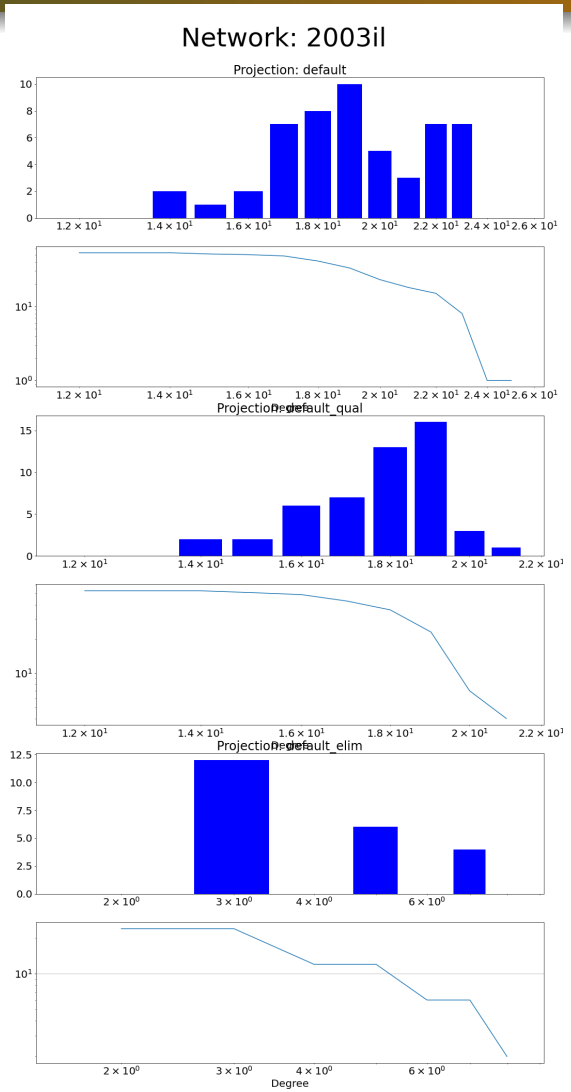
LAS

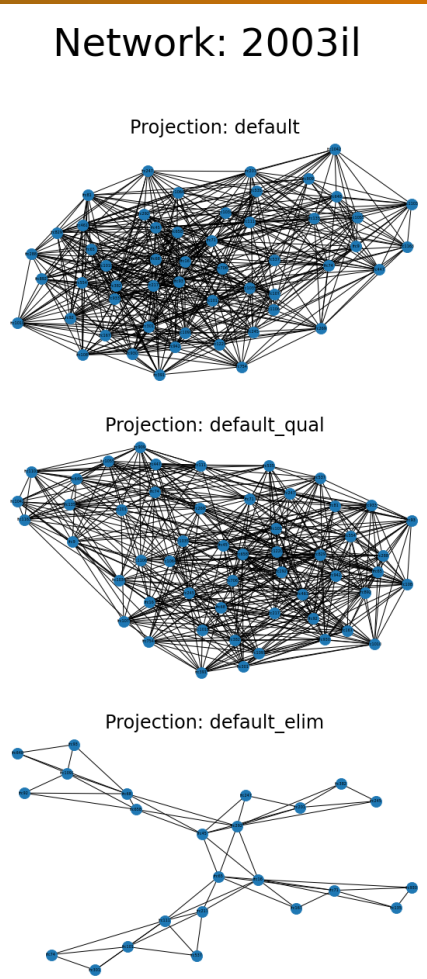
Table of Contents

2003 Midwest Regional Analysis

Pre-WI Regional: In Evanston, IL

```
Getting Projection: default
Diameter =  3
--------------------------------------------
Getting Projection: default
degree centrality
----------------------
  1. frc16 (0.5000)
  2. frc111 (0.4423)
  3. frc217 (0.4423)
  4. frc292 (0.4423)
  5. frc1091 (0.4423)
  6. frc45 (0.4423)
  7. frc658 (0.4423)
  8. frc71 (0.4423)
  9. frc101 (0.4231)
  10. frc461 (0.4231)
----------------------
Getting Projection: default
eigenvector centrality
----------------------
  1. frc16 (0.1884)
  2. frc217 (0.1677)
  3. frc292 (0.1667)
  4. frc45 (0.1647)
  5. frc658 (0.1639)
  6. frc1091 (0.1637)
  7. frc92 (0.1627)
  8. frc461 (0.1621)
  9. frc829 (0.1603)
  10. frc65 (0.1558)
----------------------
Getting Projection: default
katz centrality
----------------------
  1. frc461 (0.2297)
  2. frc92 (0.2296)
  3. frc217 (0.2144)
  4. frc1010 (0.2119)
  5. frc167 (0.1995)
  6. frc1091 (0.1979)
  7. frc16 (0.1943)
  8. frc829 (0.1929)
  9. frc65 (0.1904)
  10. frc292 (0.1891)
----------------------
```

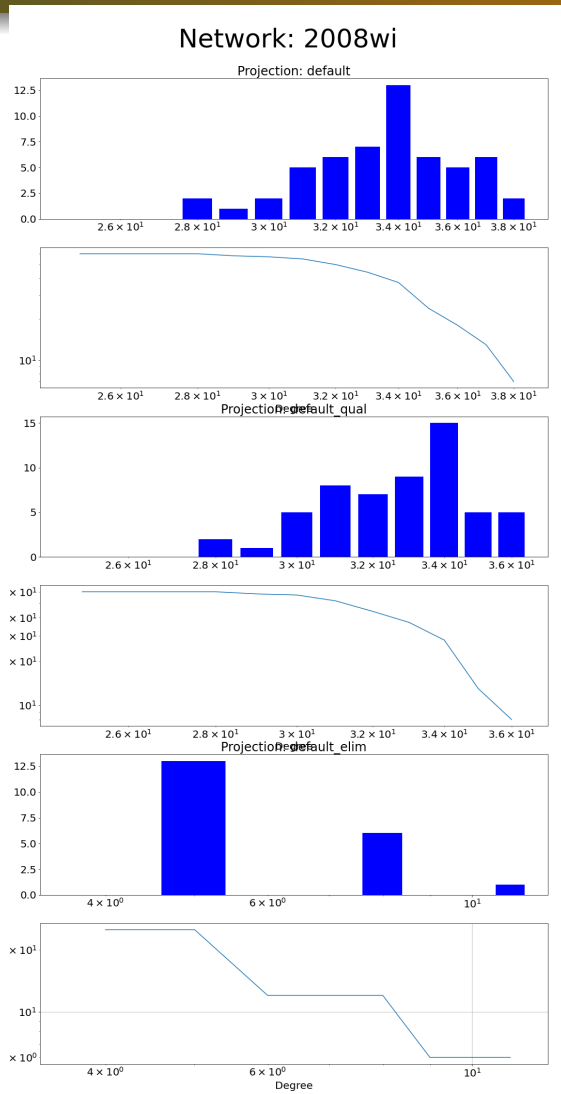


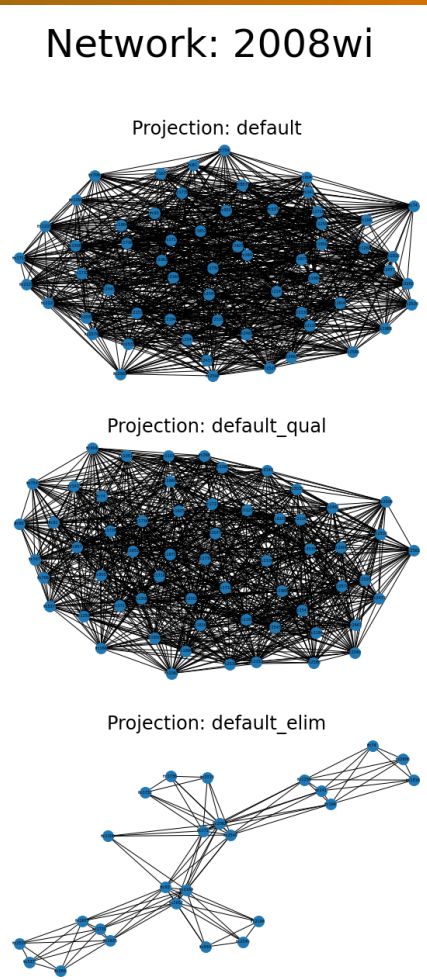


2008 Wisconsin Regional Analysis

Milwaukee, WI

```
Getting Projection: default
Diameter =  2
--------------------------------------------
Getting Projection: default
degree centrality
----------------------
  1. frc1652 (0.6610)
  2. frc171 (0.6610)
  3. frc1816 (0.6610)
  4. frc269 (0.6610)
  5. frc1675 (0.6610)
  6. frc1625 (0.6441)
  7. frc2549 (0.6441)
  8. frc1259 (0.6271)
  9. frc1306 (0.6271)
  10. frc1730 (0.6271)
----------------------
Getting Projection: default
eigenvector centrality
----------------------
  1. frc1652 (0.1478)
  2. frc171 (0.1475)
  3. frc269 (0.1475)
  4. frc1675 (0.1465)
  5. frc1816 (0.1464)
  6. frc2549 (0.1434)
  7. frc1625 (0.1430)
  8. frc1306 (0.1416)
  9. frc2169 (0.1414)
  10. frc1259 (0.1406)
----------------------
Getting Projection: default
katz centrality
----------------------
  1. frc2549 (0.2124)
  2. frc2169 (0.2025)
  3. frc2153 (0.1844)
  4. frc2574 (0.1835)
  5. frc1652 (0.1816)
  6. frc1816 (0.1809)
  7. frc1306 (0.1799)
  8. frc269 (0.1792)
  9. frc171 (0.1781)
  10. frc1675 (0.1723)
----------------------
```

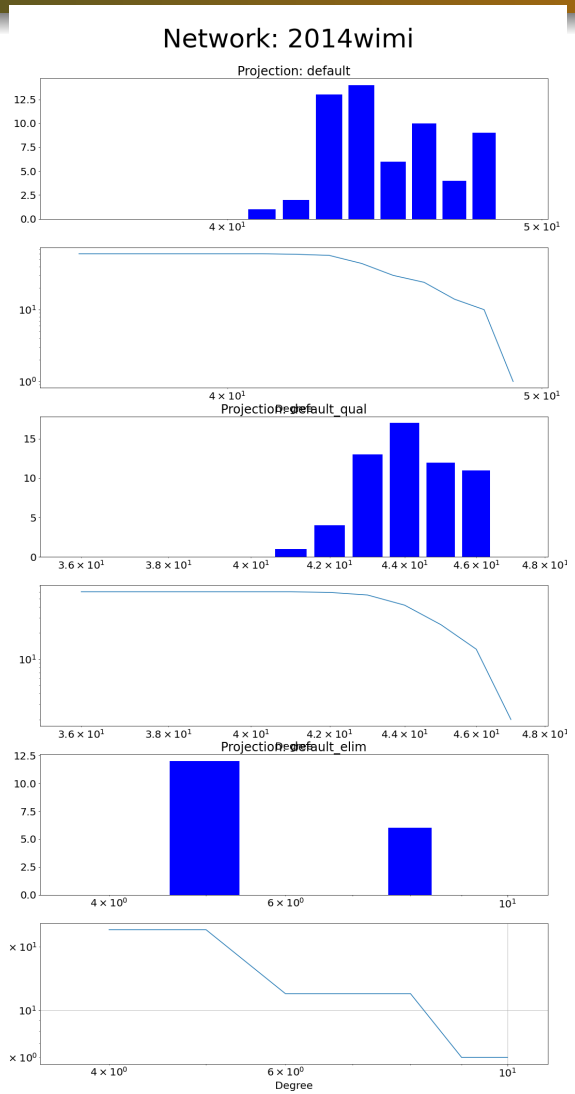


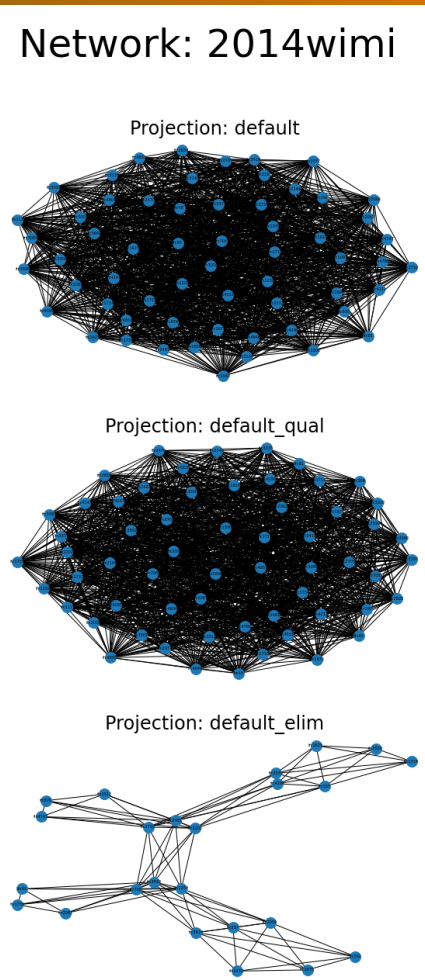


2014 Wisconsin Regional Analysis

Milwaukee, WI

```
Getting Projection: default
Diameter =  2
--------------------------------------------
Getting Projection: default
degree centrality
----------------------
  1. frc3018 (0.8475)
  2. frc1732 (0.8136)
  3. frc2039 (0.8136)
  4. frc2705 (0.8136)
  5. frc2826 (0.8136)
  6. frc3418 (0.8136)
  7. frc4296 (0.8136)
  8. frc706 (0.8136)
  9. frc2202 (0.8136)
  10. frc2194 (0.8136)
----------------------
Getting Projection: default
eigenvector centrality
----------------------
  1. frc3018 (0.1427)
  2. frc2194 (0.1383)
  3. frc4296 (0.1379)
  4. frc2039 (0.1377)
  5. frc2202 (0.1376)
  6. frc706 (0.1375)
  7. frc1732 (0.1375)
  8. frc2826 (0.1371)
  9. frc3418 (0.1371)
  10. frc2705 (0.1370)
----------------------
Getting Projection: default
katz centrality
----------------------
  1. frc3018 (0.1842)
  2. frc2194 (0.1831)
  3. frc4296 (0.1745)
  4. frc1259 (0.1691)
  5. frc706 (0.1688)
  6. frc2202 (0.1663)
  7. frc2039 (0.1661)
  8. frc1732 (0.1660)
  9. frc167 (0.1614)
  10. frc2826 (0.1603)
----------------------
```

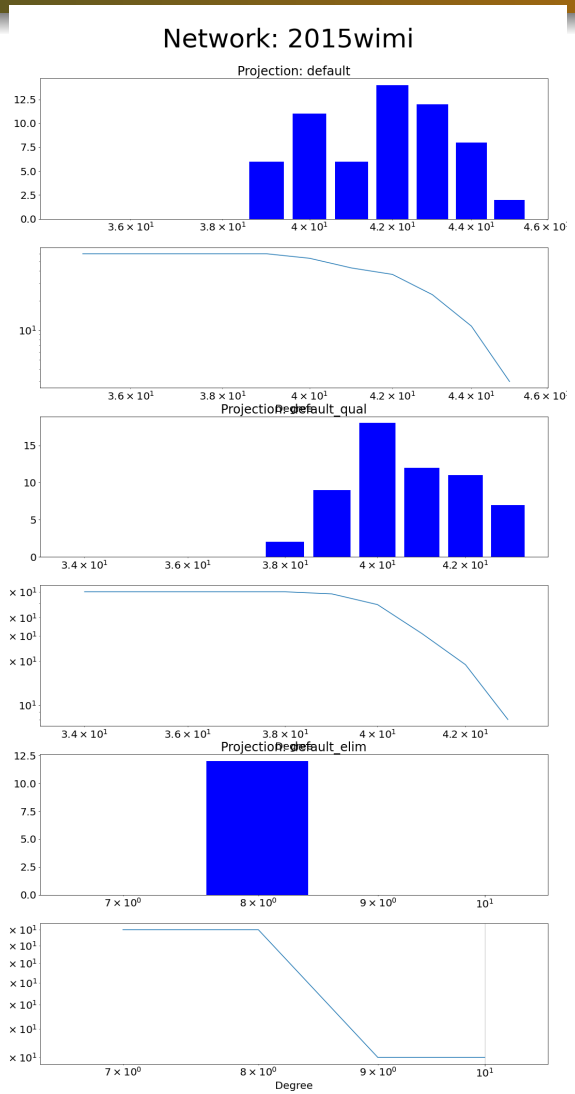


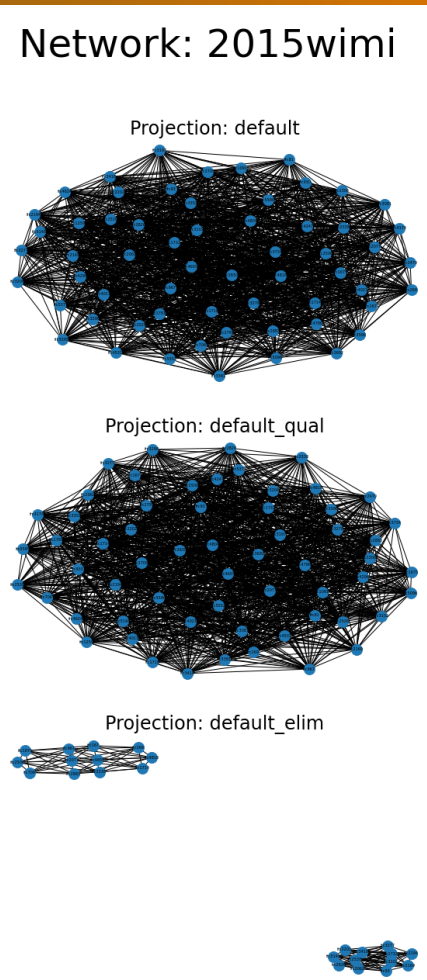


2015 Wisconsin Regional Analysis

Milwaukee, WI

```
Getting Projection: default
Diameter =  2
----------------------------------------------
Getting Projection: default
degree centrality
----------------------
  1. frc706 (0.7797)
  2. frc269 (0.7627)
  3. frc3102 (0.7627)
  4. frc1732 (0.7458)
  5. frc2077 (0.7458)
  6. frc2506 (0.7458)
  7. frc2883 (0.7458)
  8. frc4818 (0.7458)
  9. frc967 (0.7458)
  10. frc167 (0.7458)
----------------------
Getting Projection: default
eigenvector centrality
----------------------
  1. frc706 (0.1419)
  2. frc3102 (0.1389)
  3. frc269 (0.1389)
  4. frc2506 (0.1361)
  5. frc2077 (0.1358)
  6. frc1732 (0.1356)
  7. frc2883 (0.1355)
  8. frc967 (0.1355)
  9. frc167 (0.1354)
  10. frc4818 (0.1352)
----------------------
Getting Projection: default
katz centrality
----------------------
  1. frc706 (0.1864)
  2. frc3102 (0.1806)
  3. frc269 (0.1755)
  4. frc2506 (0.1650)
  5. frc2826 (0.1641)
  6. frc2077 (0.1629)
  7. frc1732 (0.1627)
  8. frc2883 (0.1602)
  9. frc1675 (0.1595)
  10. frc93 (0.1571)
----------------------
```

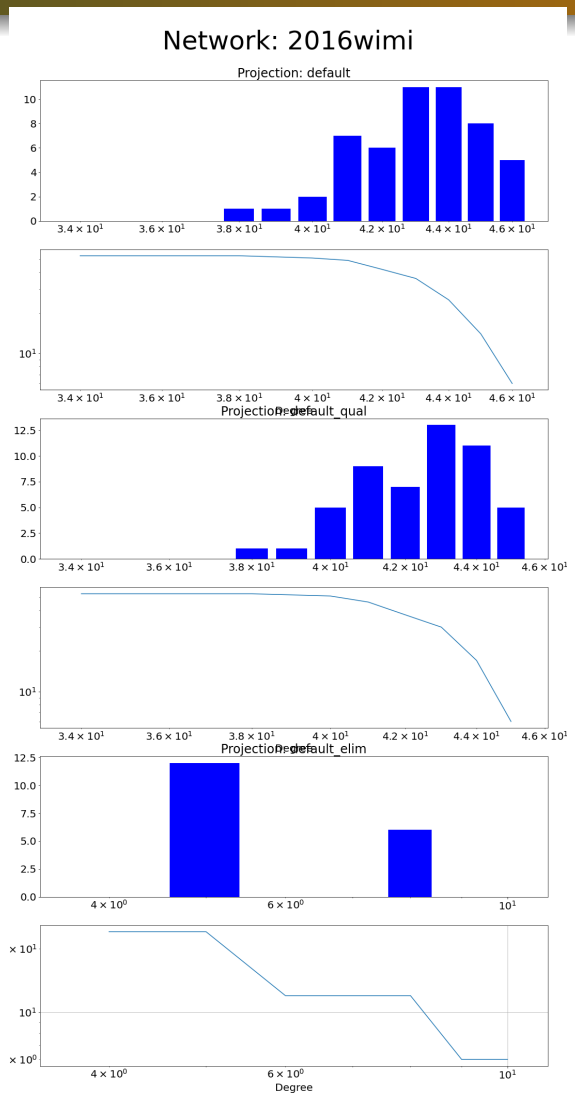


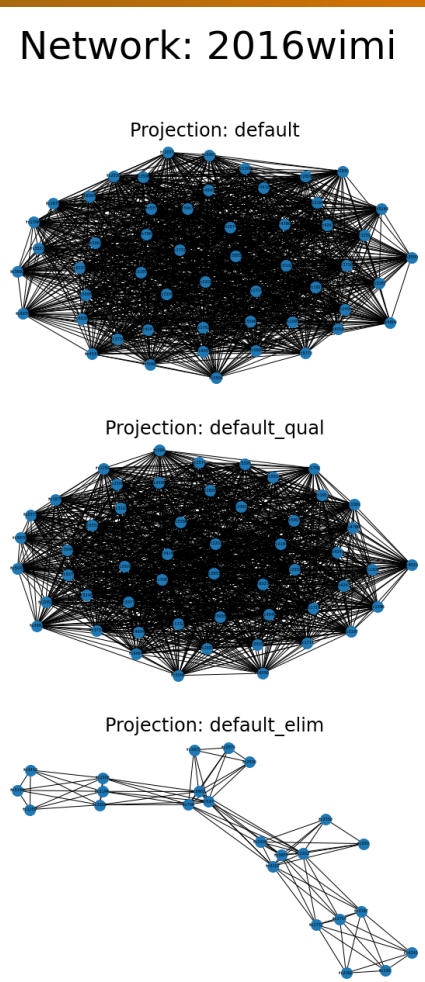


2016 Wisconsin Regional Analysis

Milwaukee, WI

```
Getting Projection: default
Diameter =  2
--------------------------------------------
Getting Projection: default
degree centrality
----------------------
  1. frc2574 (0.9038)
  2. frc1259 (0.8846)
  3. frc1306 (0.8846)
  4. frc2194 (0.8846)
  5. frc5003 (0.8846)
  6. frc2202 (0.8846)
  7. frc192 (0.8654)
  8. frc2077 (0.8654)
  9. frc2358 (0.8654)
  10. frc2530 (0.8654)
----------------------
Getting Projection: default
eigenvector centrality
----------------------
  1. frc2574 (0.1493)
  2. frc2194 (0.1463)
  3. frc1259 (0.1461)
  4. frc1306 (0.1461)
  5. frc2202 (0.1458)
  6. frc5003 (0.1457)
  7. frc2530 (0.1431)
  8. frc192 (0.1430)
  9. frc5855 (0.1429)
  10. frc2077 (0.1429)
----------------------
Getting Projection: default
katz centrality
----------------------
  1. frc2574 (0.1894)
  2. frc2194 (0.1777)
  3. frc1259 (0.1756)
  4. frc1306 (0.1739)
  5. frc2202 (0.1696)
  6. frc5003 (0.1665)
  7. frc2530 (0.1634)
  8. frc192 (0.1596)
  9. frc5855 (0.1591)
  10. frc2077 (0.1583)
----------------------
```

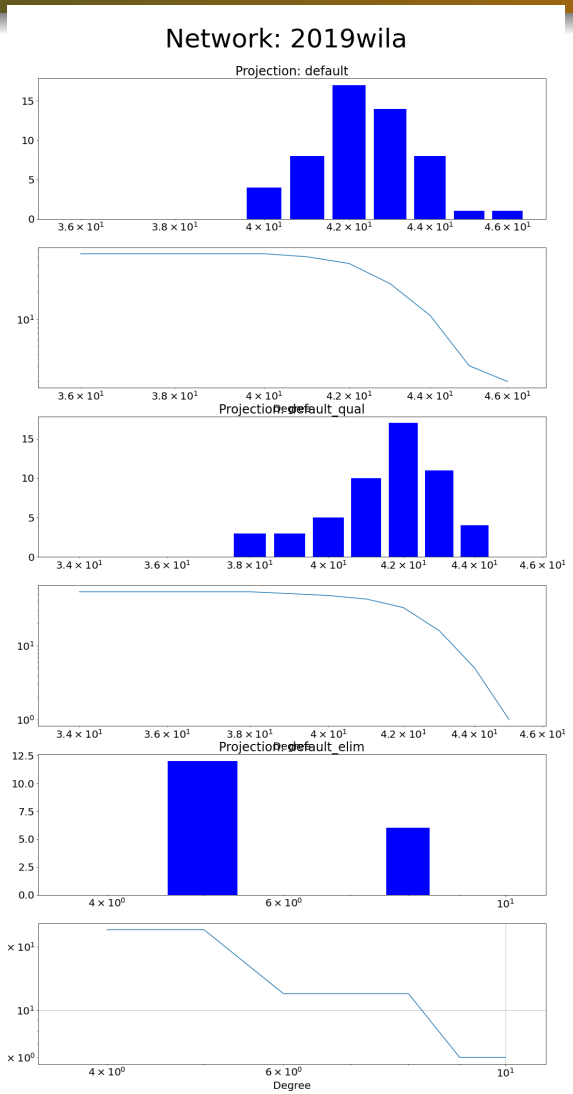




2019 Seven-Rivers Regional Analysis

La Crosse, WI

```
Getting Projection: default
Diameter =  2
-----------------------------------------------
Getting Projection: default
degree centrality
----------------------
  1. frc2081 (0.8868)
  2. frc4296 (0.8679)
  3. frc1986 (0.8491)
  4. frc3928 (0.8302)
  5. frc4786 (0.8302)
  6. frc6381 (0.8302)
  7. frc6420 (0.8302)
  8. frc7048 (0.8302)
  9. frc7596 (0.8302)
  10. frc2194 (0.8302)
----------------------
Getting Projection: default
eigenvector centrality
----------------------
  1. frc2081 (0.1502)
  2. frc4296 (0.1475)
  3. frc1986 (0.1436)
  4. frc3928 (0.1411)
  5. frc7596 (0.1411)
  6. frc6381 (0.1410)
  7. frc2194 (0.1409)
  8. frc648 (0.1407)
  9. frc4786 (0.1407)
  10. frc6420 (0.1406)
----------------------
Getting Projection: default
katz centrality
----------------------
  1. frc2081 (0.1922)
  2. frc4296 (0.1890)
  3. frc1986 (0.1613)
  4. frc6381 (0.1608)
  5. frc3928 (0.1607)
  6. frc7596 (0.1589)
  7. frc2194 (0.1572)
  8. frc648 (0.1554)
  9. frc4786 (0.1549)
  10. frc6420 (0.1520)
----------------------
```



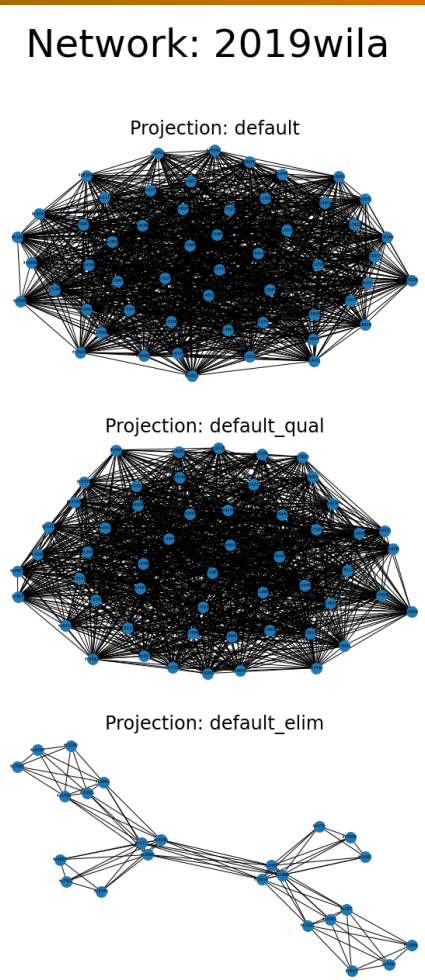
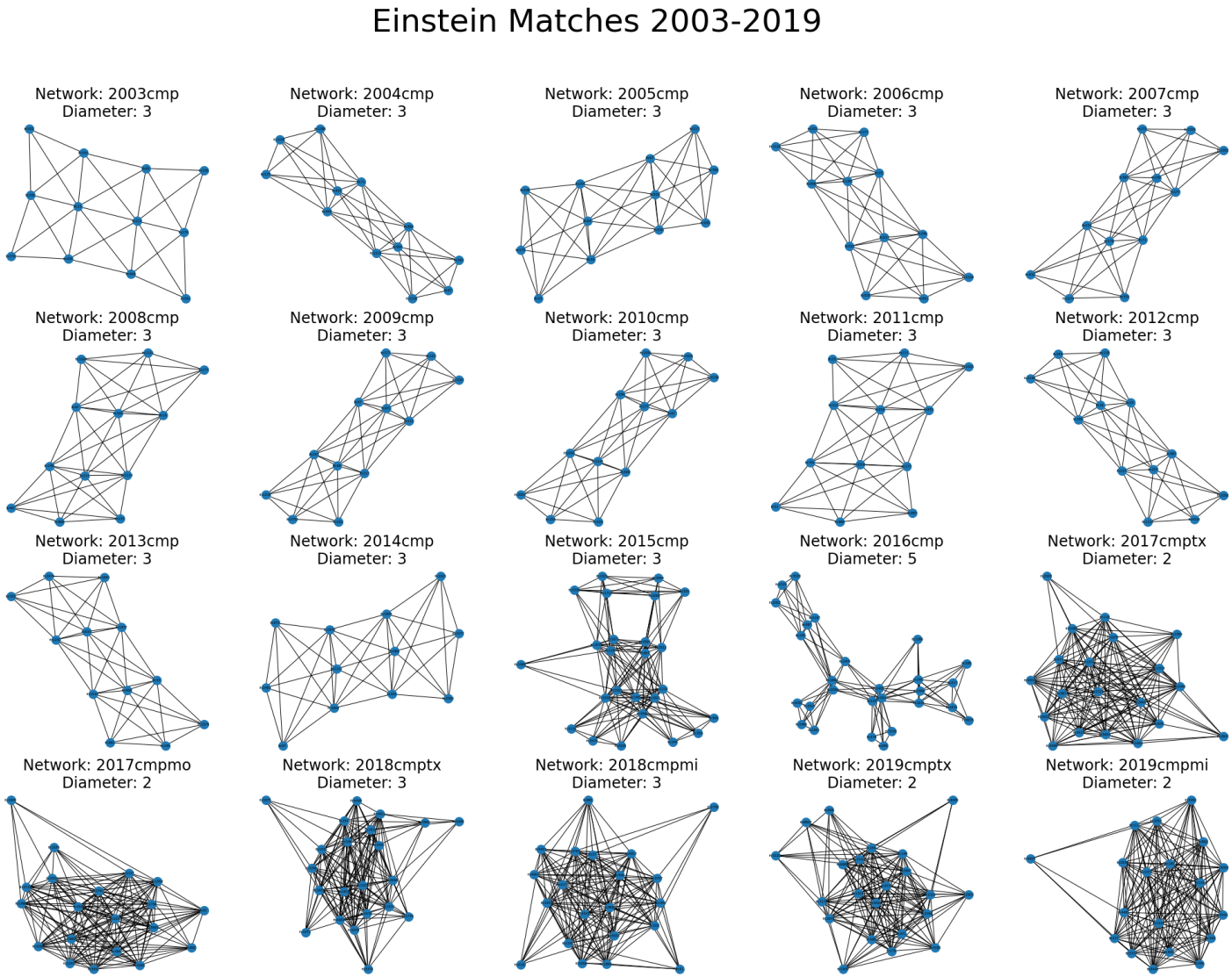


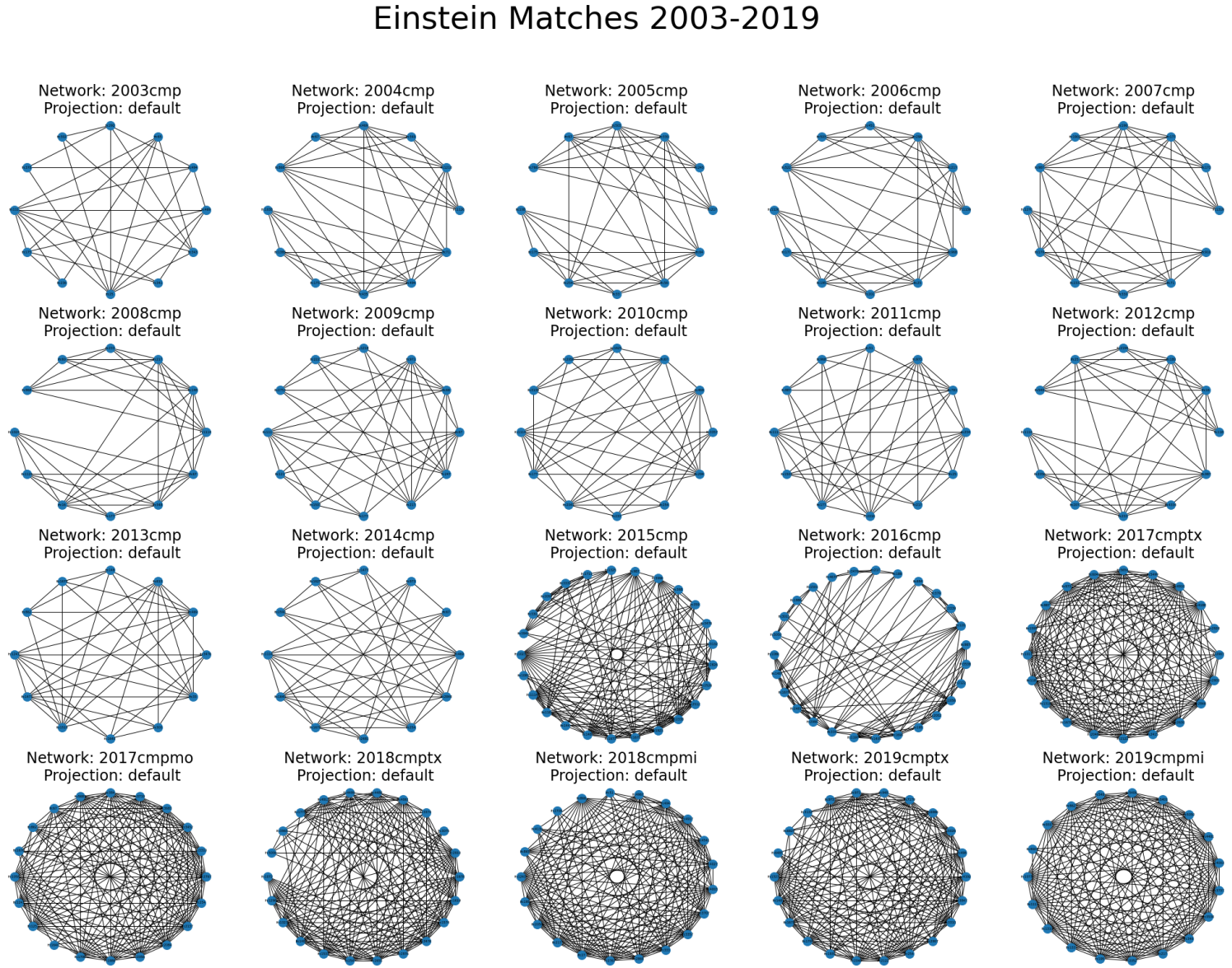
Table of Contents

Einstein (Championship) Matches



DALLAS

Einstein (Championship) Matches



LLAS

Ending

Thanks for listening...
I hope it was was interesting...
Any Questions?

Bibliography I

Bibliography isn't showing up????

- The Blue Alliance: https://www.thebluealliance.com/
- networkx

[1]