**Project Report**

2019 DataONE Summer internship program

Project 6: A Reproducible Network Analysis of the DataONE Linked Open Data graph

Intern: Audrey McCombs

Primary Mentor: Bryce Mecum

Secondary Mentor: Dave Vieglais

Date: August 2, 2019

**Introduction**

This project was implemented from June 2019 through August 2019 as part of DataONE’s summer internship program.

*Project description*

With over 800,000 datasets accessible through programmatic interfaces, DataONE provides a rich corpus of machine readable metadata that is also expressed as a linked open data (LOD) graph. The goal of this project was to explore the LOD graph of DataONE, provide a network analysis of the graph, and examine how the network differs from the content available through the traditional DataONE Application Programming Interface (API). Questions to be addressed in this project include: How interconnected are data sets and researchers? How many individual authors contributed to how many data sets? Can fields such as keywords be normalized to a small set of controlled vocabularies? How do network analysis measures differ by metadata standard, year of publication, or other facets?

*Expected outcomes*

* Identification of a set of key metrics or questions to apply to the DataONE linked open data graph
* A reproducible analysis of key metrics or questions in the form of a report that can be re-run periodically to track changes over time

*Overview*

The purpose of this project was to create reproducible code to build networks of datasets in the DataONE archives. To meet project expectations, we identified a set of key questions of interest associating the relationships among datasets in the DataONE archives to important aspects of DataONE’s mission. We identified particular statistics related to network topology that will answer those questions of interest, then created reproducible code to build the networks and calculate the relevant statistics.

Because the code is reproducible, the network can be re-built and descriptive statistics re-calculated at regular intervals. Understanding how the topological characteristics of the networks change over time can provide insight into many aspects of DataONE’s operations. Current non-network based metrics inform DataONE analysts about which member nodes are experiencing the most growth and where DataONE has been most successful in increasing the visibility of datasets. A network-based analysis allows DataONE analysts to understand the relational mechanisms that may be causing or contributing to growth and increased visibility. A better understanding of these mechanisms, provided by network topology analysis over time, can help DataONE predict and prepare for future trends in archive use.

A second goal that may be incorporated into future DataONE queries involves enhancements to DataONE’s search engine through a "suggestions" feature. Network analysis can identify communities of datasets that are related to each other through creation, contribution, and download events. Once a researcher finds a dataset in DataONE’s archives, an enhanced search engine could potentially suggest other datasets belonging to the same community, thereby enhancing the researcher’s user experience and improving the ability of the scientific community as a whole to discover, access and integrate archived datasets into current ecological and earth sciences research.

**Methods and Results**

Graph production is described in detail in the R markdown document *MakeNetwork.Rmd*, available on the DataONEorg GitHub repository “lod-graph-analysis”. That document also contains the reproducible code for creating the graphs. Data used as input and tables of statistics returned as output are also saved in the repository.

We produced three graphs:

1. ADC datasets: Arctic Data Center archive with datasets as nodes and contributors as links. This graph focused on the network structure of the datasets in the Arctic Data Center archive. Also of interest were the communities of datasets formed by common contributors.
2. ADC contributors: Arctic Data Center archive with contributors as nodes and datasets as links. This network focused on the topological characteristics of a network of individual contributors. From this graph we identified the individuals who are important to connectivity in the network, and who have contributed to a large number of datasets.
3. DataONE Smaller Archives: A subset of all DataONE archives with datasets as nodes and contributors as links. Similar to the ADC datasets graph, the focus for this network is on communities of datasets formed by query sessions. This graph also includes an external dataset attribute—archive name—that we incorporated into the network analysis.

For each graph we produced network visualizations in the software package Gephi, and calculated the graph statistics listed below using the R package igraph. (Network statistics are described for a network with datasets as nodes and users as links; for the flipped ADC contributors network, substitute “datasets” for “users” and vice versa.)

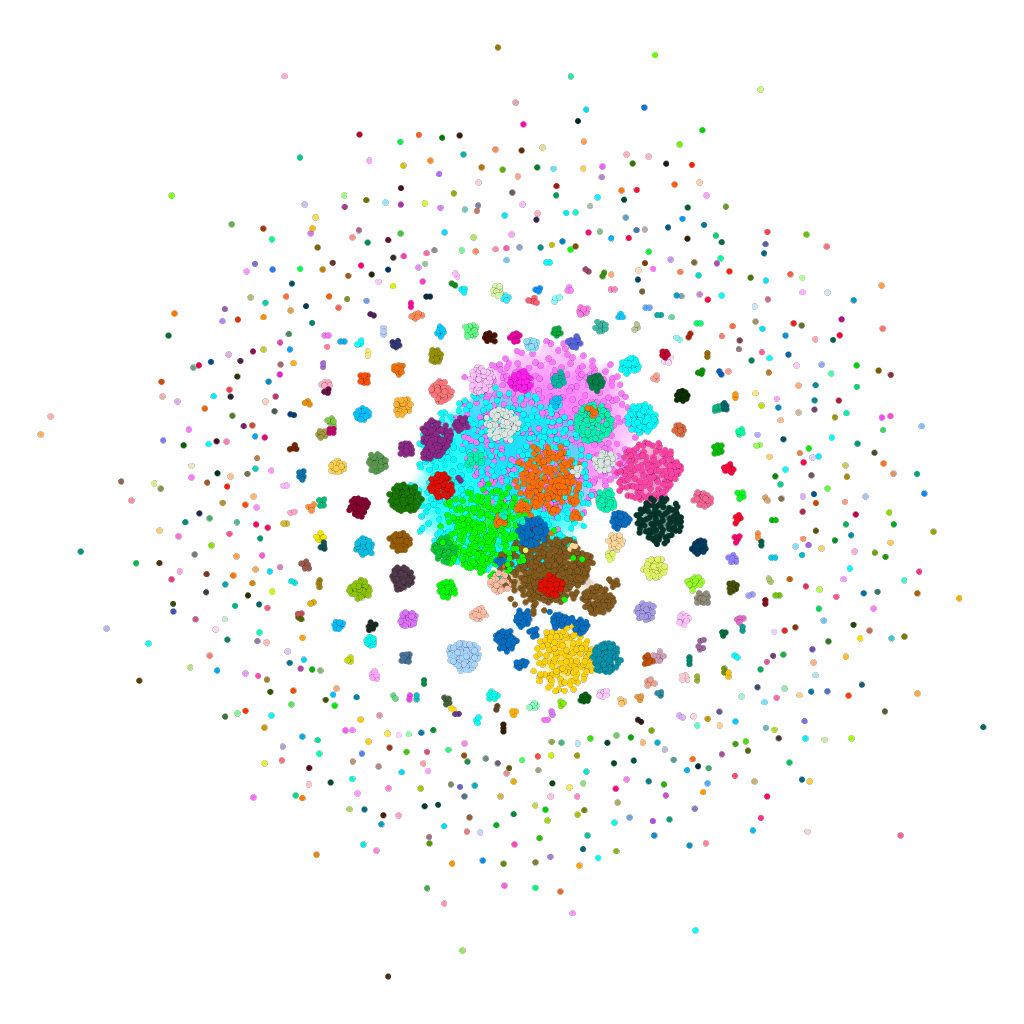
1. Number of rows in the original data table listing unique dataset-user combinations
2. Total number of unique users in the archive
3. Number of users in the archive who only interacted with one dataset
4. Number of users in the archive who interacted with more than one dataset
5. Number of interaction events; that is, the total number of times all users interacted with the datasets in the archive
6. Number of nodes (datasets) in the graph
7. Number of links (users) in the graph
8. Median degree. A node’s degree is just the number of links connected to that node. Median degree is the median number of links over all nodes in the network.
9. Mean degree. Mean number of links over all datasets in the network.
10. Max degree. Maximum node degree over all datasets in the network.
11. Number of nodes with degree one; that is, the number of datasets with only one user of the dataset.
12. Network density: how close the network is to complete. This is the ratio of realized edges to the number of possible edges. A complete network has all possible edges
13. Average shortest path length: The shortest path between any two nodes is the path between those two nodes that passes through the fewest other nodes. The length of the shortest path is the number of nodes the path passes through.
14. Network diameter: The longest shortest path on the network
15. Number of connected components: The number of components in the network that are isolated from other components.
16. Overall modularity: Modules in a network are groups of links that are more connected among themselves than they are with the rest of the network. Modules are “communities” of nodes in the network. The overall modularity score for a network measures how “clique-y” the network is, as opposed to more evenly connected throughout.

The first five statistics describe aspects of the archive as a whole, independent of the network built from datasets in the archive. Statistics 6 through 16 quantify important aspect of the topology of the entire network. Tracking how these statistics change over time can provide important information about how the archive is being used by researchers. Details about how to interpret these statistics are included in the specific network descriptions below.

In addition to these archive- and network-level statistics, we also calculated two node-level statistics of interest: 1) degree centrality, which is the degree of the node, and 2) modularity class, which is the community to which the node has been assigned by a community detection algorithm. We ran two community detection algorithms: leading eigenvector clustering and walktrap clustering. Leading eigenvector clustering works with a deterministic algorithm that conducts a spectral decomposition of a matrix related to the adjacency matrix of the network. Walktrap clustering works on a random walk algorithm and is stochastic in nature. Leading eigenvector clustering tends to find fewer communities of larger size than walktrap clustering. For the DataONE Smaller Nodes network we also included in the table of node attributes the identity of the DataONE archive that hosts the dataset.

*ADC datasets*

We created a visualization of the ADC datasets network in Gephi.



Visualization of the Arctic Data Center network of datasets. Colors

indicate communities identified by Gephi’s community detection algorithm.

In this graph visualization the colors represent modules, or communities of datasets identified by the leading eigenvector community detection algorithm described above. These are groups of datasets that are more connected with each other than they are with the rest of the graph, with connections being determined by users of the archive. The interesting topological features of this graph are captured in the network statistics listed in the table below.

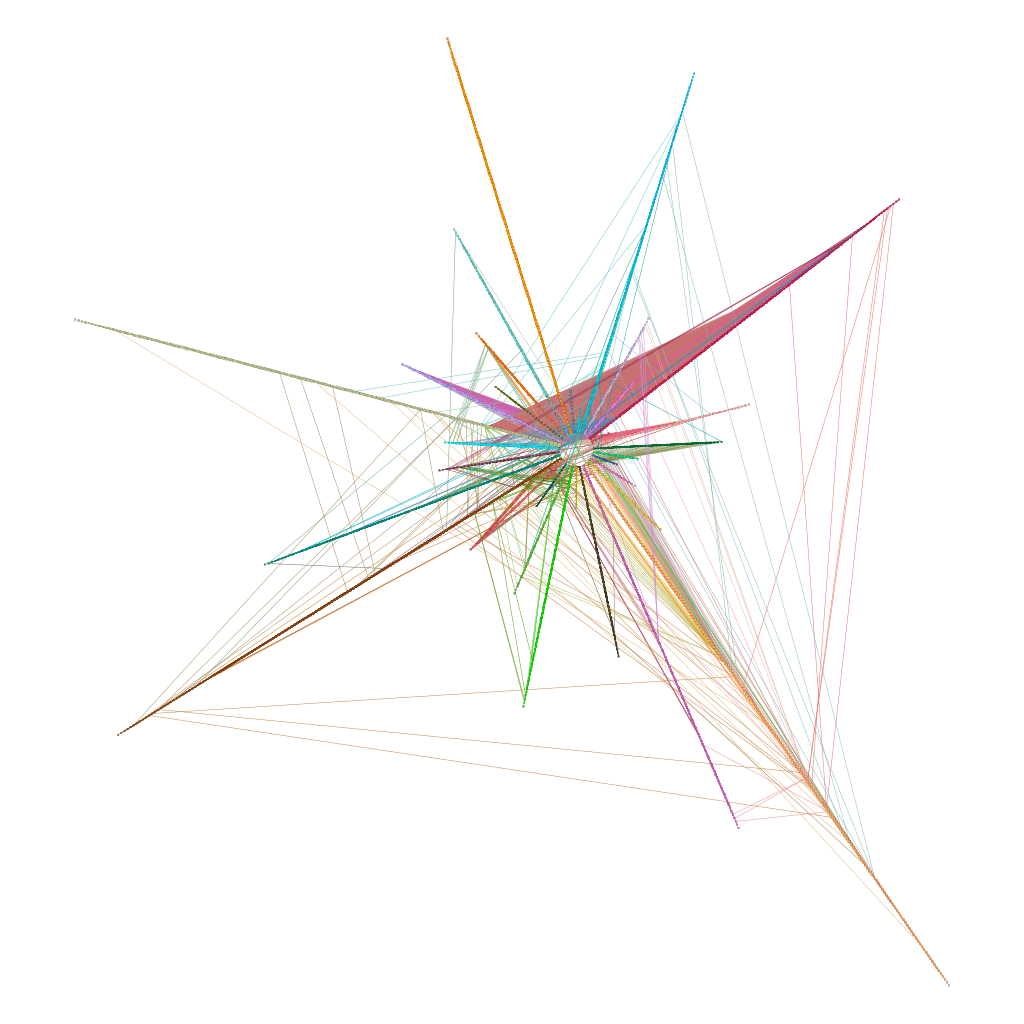
Statistics calculated for the ADC network of datasets:

|  |  |
| --- | --- |
| **Statistic** | **Value** |
| num\_rows\_csv | 9237 |
| num\_users | 2862 |
| one\_dataset\_users | 2224 |
| mult\_dataset\_users | 638 |
| interaction\_events | 6998 |
| num\_nodes | 3790 |
| num\_edges | 170719 |
| med\_degree | 35 |
| mean\_degree | 90.09 |
| max\_degree | 372 |
| num\_degree\_one | 154 |
| net\_density | 0.0238 |
| avg\_short\_path | 3.91 |
| net\_diameter | 10 |
| net\_components | 165 |
| net\_modularity | 0.7889 |

We can interpret these statistics as follows: A database query to the Arctic Data Center archive produced a table of 9,237 unique dataset-contributor pairs, with 2,862 unique contributors. Of those contributors, 2,224 worked on only one dataset, while 638 contributed to more than one dataset. The archive query captured 6,998 events in which a unique user interacted with a dataset. The network contained 3,790 nodes and 170,719 edges. The median node degree was 35 while the mean degree was about 90, indicating that the degree distribution is strongly skewed with many low values and a few very high values. The maximum number of people who contributed to a single dataset is 372. One hundred and fifty-four datasets only had one contributor. The network is only 2.38% complete, which is a low percentage and indicates the network is highly fractured—a characteristic also captured by the high number of disconnected components (165). For all the shortest paths between two nodes on the network, the average number of nodes the shortest path passes through is about 4, while the maximum number of nodes is 10. The network is highly modular, with a modularity score of 0.789, suggesting communities in the network are easily distinguished from one another.

Referring to the visualization above, these statistics capture important characteristics of the network, such as the fact that the network is highly fractured (2.38% complete) and there are many disconnected components (165 components). Many of the nodes have low degree (the nodes on the periphery of the network), while a relatively few in the middle have a high degree. In this network the communities are easy to distinguish, reflected in the network’s high modularity score of 0.789.

We can examine the central core of the network by visualizing the giant component: the fully-connected component of the graph with the highest number of nodes.



Visualization of the giant component of the Arctic Data Center network of datasets.

Colors indicate communities identified by Gephi’s community detection algorithm.

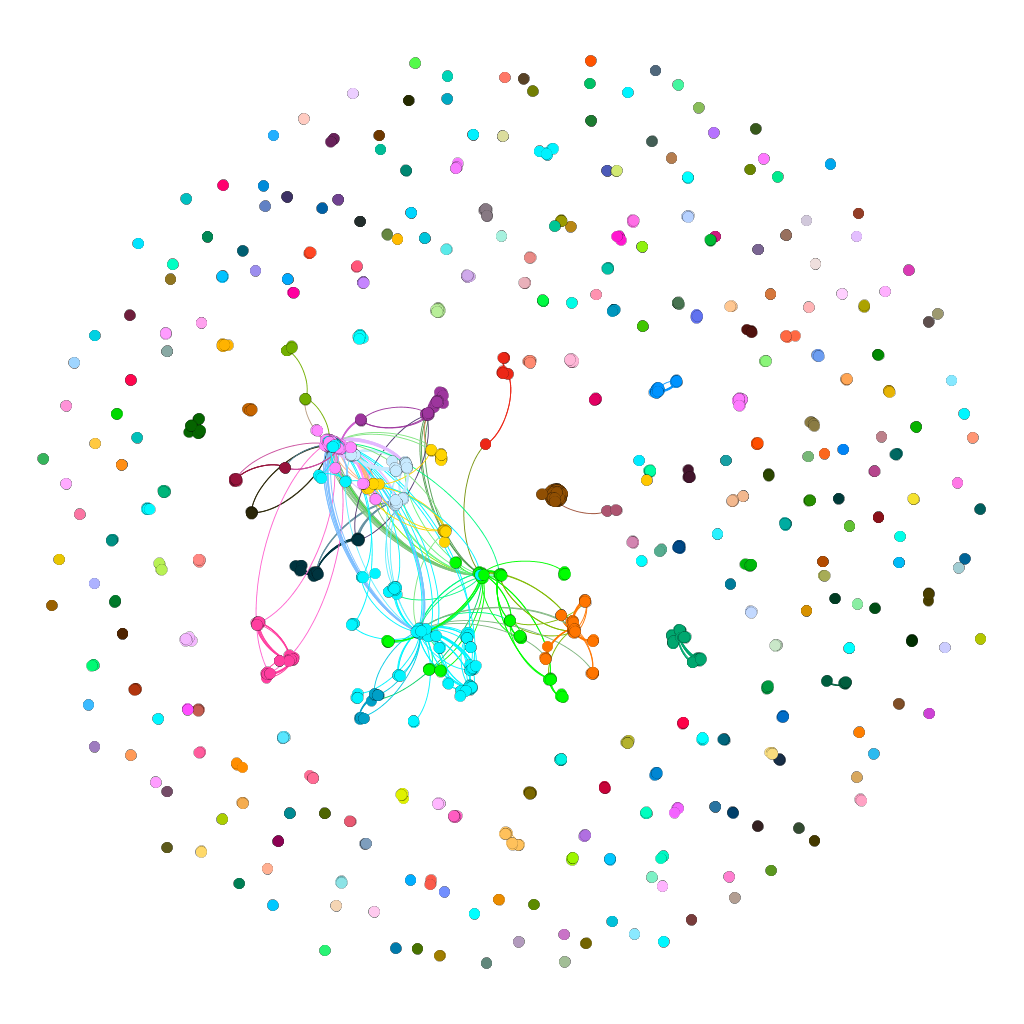
In this visualization of the giant component, colors represent communities as before (although the colors do not match the colors in the whole-graph visualization above.) For the giant component only, the community detection algorithm identified 31 communities of datasets, represented as spurs on the graph. Communities are primarily connected within themselves, however this visualization indicates that there are some connections among communities, especially between the red spur in the upper right and the green spur in the upper left.

The first fifteen rows of the node attributes table created in R for the network of datasets in the Arctic Data Center archive:

|  |  |  |  |
| --- | --- | --- | --- |
| **Dataset ID** | **Degree** | **Community identifier**  **(from eigenvalue algorithm)** | **Community identifier**  **(from random walk algorithm)** |
| doi:10.18739/A2000001N | 1 | 1 | 81 |
| doi:10.18739/A2027H | 315 | 6 | 43 |
| doi:10.18739/A2028PC7G | 109 | 8 | 62 |
| doi:10.18739/A2028W | 19 | 9 | 138 |
| doi:10.18739/A20298 | 25 | 167 | 24 |
| doi:10.18739/A2033B | 23 | 170 | 79 |
| doi:10.18739/A20531 | 315 | 6 | 43 |
| doi:10.18739/A2057CR7B | 9 | 12 | 127 |
| doi:10.18739/A2057CR99 | 109 | 8 | 62 |
| doi:10.18739/A2058X | 27 | 13 | 1 |
| doi:10.18739/A2063C | 32 | 170 | 25 |
| doi:10.18739/A20819 | 331 | 11 | 82 |
| doi:10.18739/A20B8N | 331 | 11 | 82 |
| doi:10.18739/A20C0Z | 315 | 6 | 43 |
| doi:10.18739/A20C1B | 315 | 6 | 43 |

*ADC contributors*

The network of contributors to the Arctic Data Center archive flips the previous network by setting contributors as nodes and datasets as links. The network of ADC contributors looks very similar to the network of ADC datasets, with similar topological characteristics. We produced a visualization of the contributors network:



Visualization of the Arctic Data Center network of contributors. Colors

indicate communities identified by Gephi’s community detection algorithm.

This network is much smaller than the network of datasets in the Arctic Data Center, but otherwise the two graphs share a very similar topology. The statistics for this network are listed in the table below:

|  |  |
| --- | --- |
| **Statistic** | **Value** |
| num\_rows\_csv | 9237 |
| num\_datasets | 4401 |
| one\_user\_datasets | 2478 |
| mult\_user\_datasets | 1923 |
| interaction\_events | 6744 |
| num\_nodes | 2368 |
| num\_edges | 7530 |
| med\_degree | 4 |
| mean\_degree | 6.36 |
| max\_degree | 618 |
| num\_degree\_one | 355 |
| net\_density | 0.0027 |
| avg\_short\_path | 3.37 |
| net\_diameter | 10 |
| net\_components | 282 |
| net\_modularity | 0.8132 |

We interpret these statistics as follows: This graph was produced from the same database query as the previous graph (with datasets as nodes and users as links), so the size of the database query is the same at 9,237 unique dataset-contributor pairs, with 4,401 unique datasets in the archive. Of those datasets, 2,478 had only one contributor, while 1,923 had multiple contributors. The archive query captured 6,744 events in which a dataset was accessed by a contributor. The network contained 2,368 nodes and 7,530 edges (two orders of magnitude smaller than the datasets network). The median node degree was 4 while the mean degree was about 6, indicating that the degree distribution is skewed with many low values and a few very high values. The maximum number of datasets created by an individual contributor was 618. Three hundred and fifty-five contributors only worked on one dataset. The network is only 0.27% complete, lower than the datasets network by an order of magnitude, with 282 separate components. For all the shortest paths between two nodes on the network, the average number of nodes the shortest path passes through is about 3, while the maximum number of nodes is 10. The network is highly modular, with a modularity score of 0.813, suggesting communities in the network are easily distinguished from one another.

Questions of interest for this graph involve identifying individuals who are important to connectivity in the network, and who have contributed to a large number of datasets. Betweenness centrality is a measure of how important a node is to connectivity in the network: it is proportional to the number of shortest paths involving that node. The individuals with the highest betweenness centraility on this network are:

|  |  |  |  |
| --- | --- | --- | --- |
| **Last Name** | **ORCID** | **Betweeness centrality** | **Degree** |
| Ashjian | https://orcid.org/0000-0002-7894-1519 | 80475.94 | 60 |
| O'Neel | https://orcid.org/0000-0002-9185-0144 | 65509.33 | 618 |
| Nolan |  | 63033.33 | 614 |
| Zinck |  | 63033.33 | 614 |
| Cooper |  | 48415.59 | 50 |
| Walker |  | 40165.21 | 157 |
| Grebmeier | https://orcid.org/0000-0001-7624-3568 | 26828.17 | 37 |
| Sambrotto |  | 25918.95 | 30 |
| Racine |  | 21048.4 | 53 |
| McGuire |  | 20173.34 | 61 |

Individuals who contribute to a large number of datasets are nodes with the highest degree.

|  |  |  |  |
| --- | --- | --- | --- |
| **Last Name** | **ORCID** | **Degree** | **Betweeness centrality** |
| O'Neel | https://orcid.org/0000-0002-9185-0144 | 618 | 65509.33 |
| Nolan |  | 614 | 63033.33 |
| Zinck |  | 614 | 63033.33 |
| Walker |  | 157 | 40165.21 |
| Moody |  | 106 | 8811.465 |
| Copass |  | 84 | 5575.718 |
| Knudson |  | 78 | 4341.398 |
| Epstein |  | 76 | 3250.96 |
| Chapin |  | 73 | 4144.241 |
| Beringer |  | 72 | 3644.142 |

The first twenty rows of the node attributes table created in R for the network of contributors in the Arctic Data Center archive:

|  |  |  |  |
| --- | --- | --- | --- |
| **Creator ID** | **Degree** | **Community identifier**  **(from eigenvalue algorithm)** | **Community identifier**  **(from random walk algorithm)** |
| 001496a4-2475-47eb-99bf-2406093d744d | 4 | 1 | 24 |
| 00884de9-1f56-48b9-8e0f-cfbf28ff24dd | 1 | 289 | 31 |
| 008f43e8-115c-4e77-8afa-e345ba29702a | 4 | 1 | 24 |
| 0129ac22-92a4-4d38-b201-32b0275145fd | 7 | 289 | 1 |
| 012d9ccb-e332-4ada-9c0f-d3ab34cad14f | 4 | 1 | 24 |
| 014b891b-d294-4e20-82d5-c02ef2c66196 | 50 | 289 | 6 |
| 01785d43-fe34-4d06-b05a-a1d6c11948a0 | 2 | 235 | 214 |
| 0186359b-1cde-4860-8f71-5a2375e1194c | 45 | 283 | 6 |
| 018dba33-4fd7-4739-a3e1-6081a0627071 | 3 | 3 | 6 |
| 01b1f7cd-87da-4acb-a5be-3490d5b3a87d | 4 | 1 | 24 |
| 01b93416-fa0e-45d5-ae10-016a4405ad61 | 2 | 231 | 51 |
| 01c5ea7e-a99f-43b9-b90c-cf71eb28d7d1 | 10 | 114 | 257 |
| 01d78fed-b47f-46f9-8d6c-8f867f8d3047 | 4 | 1 | 24 |
| 01ef51b5-dd82-4976-8dee-c371ca495420 | 2 | 94 | 145 |
| 01fbc405-c3e7-488e-9b72-7ed6859f85d3 | 4 | 1 | 24 |
| 0256ceaf-678e-43b8-a621-72532791604e | 3 | 263 | 49 |
| 025a206d-dbef-41fe-947e-1b29e5f75ae1 | 6 | 283 | 6 |
| 0261d004-187f-4312-9703-c0d84bce2d1f | 7 | 289 | 81 |
| 028bd679-54fb-4093-8c9b-2055650f5e9f | 4 | 1 | 24 |
| 0290c543-7225-49a2-b88f-f286cd784571 | 1 | 189 | 305 |

*DataONE Smaller Nodes*

A network for the entire DataONE archive is too large to build with the code developed for this project, so we built a network from a subset of the entire archive. Specifically, we included in the network 23 of the 29 total DataONE archive nodes, listed below.

* BCODMO
* EDI
* ESA
* ESS\_DIVE
* FEMC
* FIGSHARE\_CARY
* GOA
* GRIIDC
* IARC
* IEDA\_MGDL
* IEDA\_USAP
* mnORC1
* mnUCSB1
* mnUNM1
* NCEI
* NKN
* NMEPSCOR
* ONEShare
* PPBIO
* RW
* SEAD
* TFRI, and
* UIC

Archives that were removed from the network:

* ARCTIC
* KNB
* LTER
* PANGAEA
* R2R, and
* TERN

We built a network for the Arctic Data Center as described in the two previous sections of this report, and a network for R2R and TERN could be built in the same way. Networks for KNB, LTER, and PANGAEA are too large for the code we developed as part of this project. We calculated that the size of the edge lists for these three networks would be 104,950,604 rows for the KNB archive; 921,952,313 rows for the LTER archive; and 69,724,637 rows for the PANGAEA archive. The size of the edge list for the entire DataONE archive would be approximately 1.2 billion rows. Networks for these archives should be implemented under a distributed computing paradigm, to be developed at a future date.

To build this network of 23 DataONE archive nodes, we queried the entire DataONE archive and used session ID’s as proxies for user information, assuming that datasets queried together in the same session were related. Both visually and statistically, this network has a very similar topological structure as the networks from the Arctic Data Center: the graph is highly modular (modularity score: 0.640), fractured (272 components), and disconnected (0.54% density).

We produced two visualizations of the network of 23 archives: the first colors the nodes by modularity class while the second colors the nodes by DataONE archive ID (nodes from the BCODMO archive are all one color, nodes from the EDI archive are a different color, etc.). The image on the next page presents the network with modularity classes identified at the top, and the same network with archive identified at the bottom. A comparison of the two suggests that datasets in particular archives do form communities based on user interactions.

Network statistics calculated in R for this network of 23 DataONE archives:

|  |  |
| --- | --- |
| **Statistic** | **Value** |
| num\_users | 6717 |
| one\_dataset\_users | 3132 |
| mult\_dataset\_users | 3585 |
| interaction\_events | 57181 |
| num\_nodes | 29454 |
| num\_edges | 2342733 |
| med\_degree | 104 |
| mean\_degree | 159.08 |
| max\_degree | 890 |
| num\_degree\_one | 222 |
| net\_density | 0.0054 |
| avg\_short\_path | 5.09 |
| net\_diameter | 15 |
| net\_components | 272 |
| net\_modularity | 0.6399 |

As would be expected, this is a very large graph, with 29,454 nodes and 2,342,733 edges. We do not recommend constructing a graph with more than 1 million edges with the code developed for this project—the .csv file with the final edge list for this network was approximately 51Mb, and too large to load completely in Microsoft Excel (which has a hard limit of just over 1 million rows in a spreadsheet.) R was able to create both the edge list and the graph for this network, but working with it was slow, memory intensive, and caused R to crash several times. Visualizations in Gephi were even more difficult.

A sample of the attributes table created for this network suggests that the leading eigenvalue community detection algorithm tends to place datasets from different archives into the same community, whereas communities identified by the random walk algorithm match archive identities more consistently, although not perfectly.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Dataset ID** | **Degree** | **Community identifier**  **(from eigenvalue algorithm)** | **Community identifier**  **(from random walk algorithm)** | **Archive name** |
| 114731 | 287 | 1 | 121 | NMEPSCOR |
| 114734 | 70 | 1 | 6 | IARC |
| 114735 | 307 | 1 | 121 | mnUCSB1 |
| 114737 | 156 | 1 | 7 | NMEPSCOR |
| 114739 | 499 | 1 | 59 | NMEPSCOR |
| 114741 | 326 | 1 | 59 | NMEPSCOR |
| 114742 | 369 | 1 | 59 | NKN |
| 114743 | 132 | 1 | 7 | IARC |
| 114744 | 109 | 1 | 238 | NMEPSCOR |
| 114749 | 141 | 1 | 7 | NMEPSCOR |
| 114752 | 81 | 1 | 80 | IARC |

**Project challenges and data gaps**

The primary challenge we encountered in this project was caused by the sheer size of the networks we were working with. As noted above, networks for several of the DataONE archives are too large for standard computing in R. The R package sparklyr implements distributed computing tools using the Apache Spark analytics engine for large-scale data processing. We note that sparklyr also interfaces with GraphFrames, a graph processing library for Apache Spark that provides network analysis tools similar to igraph.

We attempted to use sparklyr while building the edge list for the larger archive networks by saving the edge list in pieces: smaller files saved in distributed memory and accessed as a whole through the Spark interface. We were not able to successfully integrate the sparklyr computing tools into our code, however. We believe that using Spark (or a similar big data analytics engine) will allow the construction and analysis of these larger networks, and that integrating our code with a big data analytics engine is an important next step for DataONE. We expect that future analysts will need to coordinate with DataONE’s information technologists and network administrators to develop a storage architecture that will allow R’s analysis tools to be applied to very large datasets.

A second challenge we encountered on this project was related to linked open data. One of the questions driving this project asked: “Can fields such as keywords be normalized to a small set of controlled vocabularies?” When we examined the keywords associated with the datasets in the Arctic Data Center, we discovered that the keywords were in different formats and used varying vocabularies. For example, the keywords field for one dataset read: “Earth Science > Physical Limnology > temperature Earth Science > Physical Limnology > specific conductance Earth Science > Physical Limnology > dissolved oxygen.” The keyword field for a different dataset read: “aquatic arctic fire carbon nitrogen alaska lake fen isotopes.” Standardizing the list of keywords, both in terms of content and format, will be time-consuming. But once the keywords are standardized they can be used to define communities in the archive networks, and different community structures defined by different community-assignment techniques can be compared.

Currently, DataONE’s training module “Lesson 7: Metadata” includes information on the proper use of keywords in metadata. Continuing to train researchers in the importance of good metadata, standardizing legacy keyword fields, and implementing a standard format for the future will all increase the usefulness of keywords in future network analysis.

**Conclusion**

The overall goal of this project was to conduct an exploratory analysis of the relational structure of datasets in various DataONE archives. To meet this goal, we created reproducible code to build and analyze three networks of DataONE datasets. We found the topology across all networks to be very similar, differing primarily in the size of the network as measured by the number of nodes and edges. The code we developed can be re-used so that archive networks can be re-built and re-analyzed over time, to create a time series of network statistics that describe how the relations among datasets, as mediated by users, change over time.

The expected outcomes of the project were met in the following ways:

1. Expected outcome: Identification of a set of key metrics or questions to apply to the DataONE linked open data graph. Methods and results: We identified sixteen network statistics to describe how users mediate the relations among datasets in an archive.
2. Expected outcome: A reproducible analysis of key metrics or questions in the form of a report that can be re-run periodically to track changes over time. Methods and results: We created reproducible code in the form of an R markdown document stored in a DataONE GitHub repository. That code builds a network from an archive query, calculates the statistics identified in step 1 above, and outputs a table of network statistics that can be tracked over time.

In addition to the expected outcomes, we created the following additional products:

1. Network analyses and visualizations of three specific networks: the Arctic Data Center network of datasets, the Arctic Data Center network of contributors, and a network of datasets from a subset of the entire DataONE archive.
2. Code to create an attributes list of community identifiers for datasets in the network. We produced this attributes list for the three networks we created. Future enhancements of DataONE’s search engine could incorporate this information into a “suggestions” feature in DataONE Search.

We identified two areas for development for this project. The first involves integrating big data computing tools into the code we produced so future analysts can work with very large networks (i.e, networks with edge lists of over 1 million rows). The second area for development involves standardizing legacy keyword fields and implementing these standards in future metadata documents, so that keywords can be incorporated into network analysis.

We believe that this project successfully developed the foundational tools for a more nuanced understanding of how DataONE archives are used by researchers. A network-based analysis will allow DataONE analysts to better understand the relational mechanisms contributing to growth and increased visibility of scientific datasets. A better understanding of these mechanisms can help DataONE predict and prepare for future trends in archive use.

For example, from the original data table that contained all pairs of user-dataset interactions in an archive, in order to build the network we needed to trim users who only interacted with one dataset. (Networks are about relations, and if a user only interacted with one dataset, then that user can’t serve as a link between a pair of nodes.) We keep track of the number of users who only interacted with one dataset in statistic #3 above. If this number decreases over time while the number of nodes in a given archive increases or stays the same, then the users of that archive are increasingly interacting with more than one dataset.

The number of nodes in a network is not necessarily the same as the number of datasets in the archive. If a dataset has only one user and that user only interacts with that one dataset, then that dataset is not included in the network for the archive. If the number of nodes in an archive’s network increases over time, then the archive is hosting an increasing number of datasets and/or datasets with more than one user . If the mean, median, or maximum degree of the network increase of time, then the archive is being used by an increasing number of researchers. A reduction in the mean or maximum shortest path length indicates that the network is becoming increasingly connected, which would suggest that users are interacting with more datasets.

This document presents reproducible code so that future analysts can at regular intervals rebuild the network and recalculate descriptive statistics.

For each network, we calculated and stored fifteen networks statistics of interest. We also ran community detection algorithms and stored community membership information. For each network, therefore, we produced two .csv files: the first is a list of networks statistics and the second is a list of node attributes, including the number of links per node and community membership identifier.

We intend that the file of network statistics will be stored and added to as the network is rebuilt and statistics recalculated over time. Each of these network statistics therefore serves as a data point in a time series, and after a suitable number of datapoints have been collected, a time-series analysis can be conducted to investigate how the network is changing over time.

The table of node attributes is intended for possible future implementation of search enhancements. If a query returns a particular dataset, the DataONE’s search algorithm can read the community identifier of that dataset and suggest to users other datasets also in the same community.

**The table of statistics**

the following network statistics:

**The table of node characteristics and community identifiers**

**A note on community detection algorithms**

For a network with datasets as nodes and contributors as links, node-level statistics aren't as informative as network-level statistics. The topological attributes of individual datasets don't answer very many questions of interest. However, two node-level statistics are worth keeping track of: degree centrality and modularity class. Degree centrality is simply the degree of the node--the number of links for that node. If a particular dataset's degree increases through time, that dataset is attracting new contributors.

Modularity class is potentially an extremely useful node-level statistic, because it can enhance search-engine functionality. If a researcher finds a particular dataset useful, other datasets in the same network community might also be useful, and DataONE's search engine might make suggestions to researchers about datasets related to the ones in their search based on modularity class.

Network scientists have developed a wide variety of community detection algorithms. In the code we produced, we used leading eigenvector clustering and walktrap clustering. Leading eigenvector clustering creates a modularity matrix based on the adjacency matrix and probability assignments, then decomposes the modularity matrix and looks for community structure based on the largest positive eigenvalue. Walktrap clustering finds communities in the network from a random walk algorithm. Leading eigenvector clustering tends to find fewer communities of larger size than walktrap clustering.

**A note on degree distributions**

**Results**

*Network statistics*

*Visualizations*

**Challenges, information gaps, and future directions**