
CyVerse Documentation

Release 1.0

CyVerse

Mar 05, 2021

Contents

1	Learning Objectives	3
2	Tutorial Maintainer(s)	5
2.1	<i>Intro to DOT language</i>	5
2.2	<i>Prepare your data</i>	6
2.3	<i>Run sfdp</i>	7
2.4	<i>Node attribute manipulation</i>	7
2.5	<i>Edge attribute manipulation</i>	8
2.6	Perform network analysis using Discovery Environment VICE app	9
3	Prerequisites	11
3.1	Downloads, access, and services	11
3.2	Platform(s)	11
3.3	Application(s) used	11
3.4	Input and example data	12



 [Learning Center Home](#)

Various types of molecular interactions are principal determinants of the system-scale behaviour of the cell. These interactions include protein–protein interaction, metabolic, signaling and transcription-regulatory networks. Recent innovations and continuing falling costs of technologies have enabled researchers to catalogue the component molecules of these networks not only at a genome-wide scale but also under a large number of different experimental conditions (e.g. time points, cell types, stimuli and treatments). In this tutorial, we will use Discovery Environment apps to analyze a sample dataset of five transcription factors that control the rate of ~14k genes.

Some experience with python programming and command-line is required to follow this tutorial. Download code and sample data for this tutorial from the [github repo](#).

CHAPTER 1

Learning Objectives

- Intro to network analysis and visualization
 - Why networks are difficult to visualize and understand
 - Powerful combination of tools and techniques to do network analysis
 - How to manipulate, slice, and dice networks in order to analyze large networks
 - Using CyVerse Discovery Environment apps to do network analysis and visualization
-

Tutorial Maintainer(s)

Who to contact if this guide needs fixing. You can also email learning@CyVerse.org

Maintainer	Institution	Contact
Iqbal Hossain	CyVerse / UA	hossain@arizona.edu
Reetu Tuteja	CyVerse / UA	reetututeja@cyverse.org



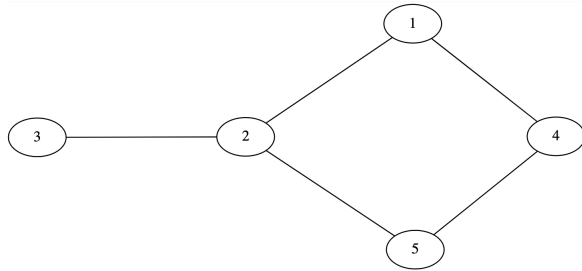
2.1 *Intro to DOT language*

DOT is the graph description language. DOT describes three main kinds of objects: graphs (or network), nodes, and edges. Nodes represent units in the network, while edges represent the connections between the nodes of the network. The node of a biological network can represent genes, proteins, mRNAs, protein/protein complexes or cellular processes. DOT graphs are typically files with the filename extension gv or dot.

DOT can be used to describe both undirected (edges with no orientations) and directed (edges with orientations) graphs. Protein–protein and genetic interactions are usually represented with an undirected network, whereas transcription factor binding, phosphorylation, and metabolic networks have directionality built into their interactions.

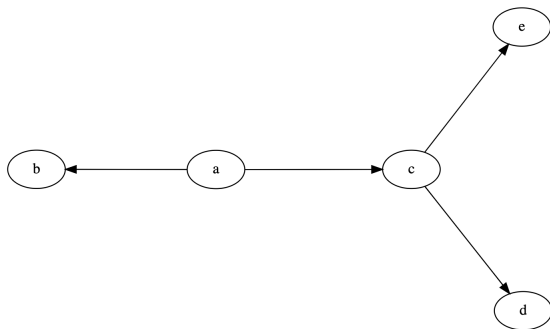
A DOT file for an undirected graph begins with the keyword `graph` followed by the name of the graph. An undirected edge between two nodes is specified using two dashes (-). Below is an example DOT file for a simple undirected graph.

```
graph graphname {
    1 -- 2;
    3 -- 2;
    4 -- 1;
    2 -- 5;
    5 -- 4;
}
```



A directed graph begins with the keyword `digraph` followed by the name of the graph. A directed edge between two nodes is specified using a dash and arrow (->). Below is an example DOT file for a simple directed graph.

```
digraph graphname {
    a -> b;
    a -> c;
    c -> d;
    c -> e;
}
```



Please check more examples DOT files [here](#)

2.2 Prepare your data

The example data for this tutorial is downloaded from the [ConnecTF database](#) that contains transcription factor (TF)-target interactions for ~ 616 TFs from three different plant species (Arabidopsis, maize and rice). The example input file is in json format, containing interactions for 5 Arabidopsis TFs (14K nodes and 23K edges). Other common interactions file formats for network visualization tools are Simple interaction file (.sif), Graph Markup Language (.gml) and Nested Network Format (.nnf).

Input Data:

Input	Description	Location
query.cyjs	Example interactions input file in cytoscape json format	iplantcollaborative > example_data > Network_analysis_webinar

Here is a sample snippet to convert interactions file in json format to dot using NetworkX Python package:

```
import networkx as nx
import os, json
from networkx.drawing.nx_agraph import write_dot

json_file = open('query.cyjs')
data = json.load(json_file)

G = nx.Graph()
for n in data['elements']['nodes']:
    ntype = n['data']['type']
    if n['data']['type'] == "": ntype = "None"
    G.add_node(n['data']['id'], type=ntype, label=n['data']['name'])

for n in data['elements']['edges']:
    G.add_edge(n['data']['source'], n['data']['target'])

print(nx.info(G))
out_file_name="network.dot"
write_dot(G, out_file_name)
```

2.3 Run sfdp

Scalable Force Directed Placement (sfdp) algorithm is part of Graphviz software. It's a fast multilevel force directed algorithm that efficiently layout very large graphs in a reasonably short time. Check more about sfdp and Graphviz [here](#).

Here is the sfdp command to create layout for dot file generated in the previous step:

```
sfdp -Goverlap=prism -Nshape=point -Goutputorder=edgesfirst -Tsvg network.dot -O
```

2.4 Node attribute manipulation

NetworkX library allows to attach attributes such as weight, labels, color to networks, nodes or edges. Attributes are provided in key/value pairs. Check more about adding attributes using NetworkX [here](#).

Here is a snippet adding node colors to network generated in previous step:

```
import networkx as nx
import os
from networkx.drawing.nx_agraph import write_dot
import pygraphviz as pgv

G=nx.Graph(pgv.AGraph("network.dot"))
```

(continues on next page)

(continued from previous page)

```

color={"METABOLIC":"red", "OTHER_RNA":"green", "TXNFACTOR":"blue", "PRE_TRNA":"yellow"}
#Set color on node based on node type
for n in G.nodes():
    if G.nodes[n]['type'] in color:
        G.nodes[n]['color']=color[G.nodes[n]['type']]
    else:
        G.nodes[n]['color']='black'

out_file_name="network3.dot"
write_dot(G, out_file_name)
os.system("sfdp -Goverlap=prism -Nshape=point -Goutputorder=edgesfirst -Tsvg "+out_
↪file_name+" -O")

```

2.5 Edge attribute manipulation

Here is an example to manipulate edge attributes:

```

import networkx as nx
from networkx.drawing.nx_agraph import write_dot
import pygraphviz as pgv
import os

G=nx.Graph(pgv.AGraph("network.dot"))


color={"METABOLIC":"red", "OTHER_RNA":"green", "TXNFACTOR":"blue", "PRE_TRNA":"yellow"}

for n in G.nodes():
    if G.nodes[n]['type'] in color:
        G.nodes[n]['color']=color[G.nodes[n]['type']]
    else:
        G.nodes[n]['color']='black'
    G.nodes[n]['width']=min( G.degree(n)/15, 1.5)
#set edge color same as node if the both terminal same type
for e in G.edges():
    if G.nodes[e[0]]['type'] == G.nodes[e[1]]['type']:
        G.edges[e]['color']=G.nodes[e[0]]['color']

out_file_name="network5.dot"
write_dot(G, out_file_name)
os.system("sfdp -Goverlap=prism -Nshape=point -Goutputorder=edgesfirst -Tsvg "+out_
↪file_name+" -O")

```

Fix or improve this documentation

- Search for an answer:
- Ask us for help: click  on the lower right-hand side of the page
- Report an issue or submit a change:
- Send feedback: learning@CyVerse.org



2.6 Perform network analysis using Discovery Environment VICE app

1. Launch the Jupyter Lab Graphviz app

After you login to DE, open the Apps window and search for Jupyter Lab with key word *Jupyterlab-GraphViz* or launch the app with webinar example scripts and data Powered By CyVerse

2. Navigate to the JupyterLab url

Unlike regular DE apps once the analysis starts running you will get an url. Click on your notifications, and then by clicking on the “Access your running Analysis here” url. You will be automatically redirected to the app after it finishes the setting phase.

The video tutorial guides how to use the graphviz VICE app in discovery environment.

Fix or improve this documentation

- Search for an answer:
- Ask us for help: click on the lower right-hand side of the page
- Report an issue or submit a change:
- Send feedback: learning@CyVerse.org



CHAPTER 3

Prerequisites

3.1 Downloads, access, and services

In order to complete this tutorial you will need access to the following services/software

Prerequisite	Preparation/Notes	Link/Download
CyVerse account	You will need a CyVerse account to complete this exercise	
Cyberduck	Standalone software for upload/download to Data Store	

3.2 Platform(s)

We will use the following CyVerse platform(s):

Platform	Interface	Link	Platform Tour
Data Store	GUI/Command line		
Discovery Environment	Web/Point-and-click		

3.3 Application(s) used

Discovery Environment App(s):

App name	Version	Description	App link	Notes/other links
Jupyterlab-GraphViz	1.0	JupyterLab with Graphviz package		

3.4 Input and example data

In order to complete this tutorial you will need to have the following inputs prepared

Input File(s)	Format	Preparation/Notes	Example Data
query.cyjs	Cytoscape json	Data downloaded from ConneCTF	iplantcollaborative > example_data > Network_analysis_webinar

Fix or improve this documentation

- Search for an answer:
- Ask us for help: click  on the lower right-hand side of the page
- Report an issue or submit a change:
- Send feedback: learning@CyVerse.org

 [Learning Center Home](#)