project

July 23, 2021

Name: Julian Arts
Date: 23-07-2021
System: Ubuntu 18.04.5 LTS (GNU/Linux 5.4.0-64-generic x86_64)

[1]:
```
# Installing the packages for cytoscape analysis
import biomart
from biomart import BiomartServer
import pandas as pd
import ipycytoscape
import os
import re
import ipywidgets as widgets
import networkx as nx
from ipycytoscape import *
from networkx.readwrite import json_graph
import json
```

[2]:
```
# Importing the variant text file with rsnumbers
try:
    rsdata = pd.read_csv('variant_list.txt',header = None)
except: print("pandas.MissingDataError: Provided file is empty or path to file␣
 ↪does not exist")
pass

# Checking if the data import worked
rsdata
```

[2]:

	0
0	rs61748411
1	rs61751443
2	rs28935168
3	rs61751449
4	rs63750264
5	rs11136000
6	rs2075650
7	rs6656401
8	rs4147929
9	rs157580

```
10      rs6701713
11      rs9349407
12      rs5030858
13      rs5030849
14     rs62508646
15      rs5030856
16    rs372915038
17    rs375378714
18    rs148157138
19    rs141088742
20    rs374673901
21    rs150277632
22    rs368707795
23    rs139237860
24  rs143223844\t
25    rs147154860
26    rs369673538
```

[3]:
```
# Naming the column in the pandas dataframe as "rsnumbers"
rsdata.names = "rsnumbers"
rsdata_named = rsdata.rename(columns={0: 'rsnames'})

# Removing unwanted characters after numbers in rsnames
rsdata_named = rsdata_named.replace(to_replace ='\D+$', value = '', regex =␣
 ↪True)
rsdata_named
```

[3]:
```
        rsnames
0    rs61748411
1    rs61751443
2    rs28935168
3    rs61751449
4    rs63750264
5    rs11136000
6     rs2075650
7     rs6656401
8     rs4147929
9      rs157580
10    rs6701713
11    rs9349407
12    rs5030858
13    rs5030849
14   rs62508646
15    rs5030856
16  rs372915038
17  rs375378714
18  rs148157138
```

```
19  rs141088742
20  rs374673901
21  rs150277632
22  rs368707795
23  rs139237860
24  rs143223844
25  rs147154860
26  rs369673538
```

[4]:
```
# Using the Biomart API (based on documentation: "https://pypi.org/project/
 ↪biomart/")
server = BiomartServer("http://www.ensembl.org/biomart")
server.verbose = True  # provides setting up details
new_list = []
# Select dataset to check against, to speed up use of Biomart
hs_snp = server.datasets['hsapiens_snp']
hs_snp
```

```
[BiomartServer:'http://www.ensembl.org/biomart/martservice'] Fetching datasets
[BiomartServer:'http://www.ensembl.org/biomart/martservice'] Fetching databases
[BiomartDatabase:'Ensembl Genes 104'] Fetching datasets
[BiomartDatabase:'Mouse strains 104'] Fetching datasets
[BiomartDatabase:'Sequence'] Fetching datasets
[BiomartDatabase:'Ontology'] Fetching datasets
[BiomartDatabase:'Genomic features 104'] Fetching datasets
[BiomartDatabase:'Ensembl Variation 104'] Fetching datasets
[BiomartDatabase:'Ensembl Regulation 104'] Fetching datasets
```

[4]:
```
Human Short Variants (SNPs and indels excluding flagged variants) (GRCh38.p13)
```

[5]:
```
# Generating a list from the rsdata dataframe
convlst = []
for rsnum in rsdata_named["rsnames"]:
    convlst.append(rsnum)
convlst
```

[5]:
```
['rs61748411',
 'rs61751443',
 'rs28935168',
 'rs61751449',
 'rs63750264',
 'rs11136000',
 'rs2075650',
 'rs6656401',
 'rs4147929',
 'rs157580',
 'rs6701713',
```

```
 'rs9349407',
 'rs5030858',
 'rs5030849',
 'rs62508646',
 'rs5030856',
 'rs372915038',
 'rs375378714',
 'rs148157138',
 'rs141088742',
 'rs374673901',
 'rs150277632',
 'rs368707795',
 'rs139237860',
 'rs143223844',
 'rs147154860',
 'rs369673538']
```

[6]:
```
# Querying the rsnumbers to find the ensembl_gene_name
response = hs_snp.search({'filters': {'snp_filter': convlst }, 'attributes': ␣
 ↪[u'refsnp_id',u'ensembl_gene_name']})
```

```
[BiomartDataset:'hsapiens_snp'] Searching using following params:
{'attributes': ['refsnp_id', 'ensembl_gene_name'],
 'filters': {'snp_filter': ['rs61748411',
                            'rs61751443',
                            'rs28935168',
                            'rs61751449',
                            'rs63750264',
                            'rs11136000',
                            'rs2075650',
                            'rs6656401',
                            'rs4147929',
                            'rs157580',
                            'rs6701713',
                            'rs9349407',
                            'rs5030858',
                            'rs5030849',
                            'rs62508646',
                            'rs5030856',
                            'rs372915038',
                            'rs375378714',
                            'rs148157138',
                            'rs141088742',
                            'rs374673901',
                            'rs150277632',
                            'rs368707795',
                            'rs139237860',
```

```
                                'rs143223844',
                                'rs147154860',
                                'rs369673538']}}
[BiomartDataset:'hsapiens_snp'] Fetching filters
[BiomartDataset:'hsapiens_snp'] Fetching attributes
[BiomartDataset] search query:
b'<Query virtualSchemaName="default" formatter="TSV" header="0" uniqueRows="1"
datasetConfigVersion="0.6" count=""><Dataset name="hsapiens_snp"
interface="default"><Filter name="snp_filter" value="rs61748411,rs61751443,rs289
35168,rs61751449,rs63750264,rs11136000,rs2075650,rs6656401,rs4147929,rs157580,rs
6701713,rs9349407,rs5030858,rs5030849,rs62508646,rs5030856,rs372915038,rs3753787
14,rs148157138,rs141088742,rs374673901,rs150277632,rs368707795,rs139237860,rs143
223844,rs147154860,rs369673538" /><Attribute name="refsnp_id" /><Attribute
name="ensembl_gene_name" /></Dataset></Query>'
```

[7]:
```
# Decode the response and generate a list with rsnumbers and ensembl gene IDs
convlst2 = [["rsnames","gene_id"]]
for line in response.iter_lines():
    line = line.decode('utf-8')
    convlst2.append(line.split("\t"))
convlst2
```

[7]:
```
[['rsnames', 'gene_id'],
 ['rs11136000', 'ENSG00000120885'],
 ['rs139237860', 'ENSG00000176165'],
 ['rs141088742', 'ENSG00000176165'],
 ['rs143223844', 'ENSG00000176165'],
 ['rs147154860', 'ENSG00000176165'],
 ['rs148157138', 'ENSG00000176165'],
 ['rs150277632', 'ENSG00000176165'],
 ['rs157580', 'ENSG00000130204'],
 ['rs2075650', 'ENSG00000130204'],
 ['rs28935168', 'ENSG00000169057'],
 ['rs368707795', 'ENSG00000176165'],
 ['rs369673538', 'ENSG00000176165'],
 ['rs372915038', 'ENSG00000176165'],
 ['rs374673901', 'ENSG00000176165'],
 ['rs375378714', 'ENSG00000176165'],
 ['rs4147929', 'ENSG00000064687'],
 ['rs5030849', 'ENSG00000171759'],
 ['rs5030856', 'ENSG00000171759'],
 ['rs5030858', 'ENSG00000171759'],
 ['rs61748411', 'ENSG00000169057'],
 ['rs61751443', 'ENSG00000169057'],
 ['rs61751449', 'ENSG00000169057'],
 ['rs62508646', 'ENSG00000171759'],
 ['rs63750264', 'ENSG00000142192'],
```

```
 ['rs6656401', 'ENSG00000203710'],
 ['rs6701713', 'ENSG00000203710'],
 ['rs9349407', 'ENSG00000198087']]
```

[8]:
```
# Generate a pandas dataframe from the list
convdf = pd.DataFrame(convlst2[1:], columns=convlst2[0])
convdf["id"] = convlst
convdf
```

[8]:

	rsnames	gene_id	id
0	rs11136000	ENSG00000120885	rs61748411
1	rs139237860	ENSG00000176165	rs61751443
2	rs141088742	ENSG00000176165	rs28935168
3	rs143223844	ENSG00000176165	rs61751449
4	rs147154860	ENSG00000176165	rs63750264
5	rs148157138	ENSG00000176165	rs11136000
6	rs150277632	ENSG00000176165	rs2075650
7	rs157580	ENSG00000130204	rs6656401
8	rs2075650	ENSG00000130204	rs4147929
9	rs28935168	ENSG00000169057	rs157580
10	rs368707795	ENSG00000176165	rs6701713
11	rs369673538	ENSG00000176165	rs9349407
12	rs372915038	ENSG00000176165	rs5030858
13	rs374673901	ENSG00000176165	rs5030849
14	rs375378714	ENSG00000176165	rs62508646
15	rs4147929	ENSG00000064687	rs5030856
16	rs5030849	ENSG00000171759	rs372915038
17	rs5030856	ENSG00000171759	rs375378714
18	rs5030858	ENSG00000171759	rs148157138
19	rs61748411	ENSG00000169057	rs141088742
20	rs61751443	ENSG00000169057	rs374673901
21	rs61751449	ENSG00000169057	rs150277632
22	rs62508646	ENSG00000171759	rs368707795
23	rs63750264	ENSG00000142192	rs139237860
24	rs6656401	ENSG00000203710	rs143223844
25	rs6701713	ENSG00000203710	rs147154860
26	rs9349407	ENSG00000198087	rs369673538

[9]:
```
# Generate a dataframe with the edges: rsnumbers to ensembl gene IDs
edges = pd.DataFrame(
    {
        "source": convdf["rsnames"],
        "target": convdf["gene_id"],
        "my_edge_key": [1] * 27,
        "weight": [1] * 27,
    }
)
```

[10]:
```
# check the edges dataframe
edges
```

[10]:

	source	target	my_edge_key	weight
0	rs11136000	ENSG00000120885	1	1
1	rs139237860	ENSG00000176165	1	1
2	rs141088742	ENSG00000176165	1	1
3	rs143223844	ENSG00000176165	1	1
4	rs147154860	ENSG00000176165	1	1
5	rs148157138	ENSG00000176165	1	1
6	rs150277632	ENSG00000176165	1	1
7	rs157580	ENSG00000130204	1	1
8	rs2075650	ENSG00000130204	1	1
9	rs28935168	ENSG00000169057	1	1
10	rs368707795	ENSG00000176165	1	1
11	rs369673538	ENSG00000176165	1	1
12	rs372915038	ENSG00000176165	1	1
13	rs374673901	ENSG00000176165	1	1
14	rs375378714	ENSG00000176165	1	1
15	rs4147929	ENSG00000064687	1	1
16	rs5030849	ENSG00000171759	1	1
17	rs5030856	ENSG00000171759	1	1
18	rs5030858	ENSG00000171759	1	1
19	rs61748411	ENSG00000169057	1	1
20	rs61751443	ENSG00000169057	1	1
21	rs61751449	ENSG00000169057	1	1
22	rs62508646	ENSG00000171759	1	1
23	rs63750264	ENSG00000142192	1	1
24	rs6656401	ENSG00000203710	1	1
25	rs6701713	ENSG00000203710	1	1
26	rs9349407	ENSG00000198087	1	1

[11]:
```
# Convert the edges dataframe to json format where the names are added
js = convdf.to_json(orient = 'records')
js2 = json.loads(js)

js3 = edges.to_json(orient = 'records')
js4 = json.loads(js3)

# Generate an empty dictionary
d = {}

# Adding the id to the dictionary
d["id"] = convlst

# Generate the dictonary as json format
d["nodes"] = js2
```

```
d["edges"] = js4

# Plot the interactive cytoscape network in jupyter notebook
undirected = ipycytoscape.CytoscapeWidget()
undirected.set_tooltip_source("name")
undirected.graph.add_graph_from_json(d)

undirected.set_style([                        {
                        'selector': 'node[id]',
                             'style': {
                                  'font-family': 'helvetica',
                                  'font-size': '20px',
                                  'label': 'data(id)'}
                        }])
display(undirected)
```

```
CytoscapeWidget(cytoscape_layout={'name': 'cola'}, cytoscape_style=[{'selector':␣
↪'node[id]', 'style': {'font-f…
```

[12]:
```
# Export to csv for importing the network to Cytoscape, linking the pathways to␣
↪the ensembl gene IDs
convdf.to_csv('edges.csv', index=False)
```