Cleaning Products and Ingredient Network

Introduction

The code in this document is based on the paper "Flavor network and the principles of food pairing" by Yong-Yeol Ahn, Sebastian E. Ahnert, James P. Bagrow & Albert-László Barabási [1]. The paper analyzes the ingredients used in recipes from various cuisines in order to determine which ingredients mix well together to form recipes, as well as find the authentic ingredients for the analyzed cuisines.

In our study, we use similar approaches to find out the chemical ingredients used in cleaning products from 12 different areas in both bathrooms and kitchens, in order to determine which ingredients are more often mixed used, and what related health outcomes there are.

Figure 1 Bipartite Network

We connected each cleaning products to its chemical ingredients.

- cleaning products listed were selected based on the top 70% usage.
- chemical ingridents were selected with either hazardous or above a reportable level as defined by OSHA 29 CFR 1910.1200.

Connections

```
import pandas as pd
import networkx as nx
from networkx.algorithms import bipartite
# basic graph drawing capability
%matplotlib inline
import matplotlib.pyplot as plt
```

```
Figure1A = pd.read_excel('FigureData.xlsx', sheet_name='Figure1A_total')
#Figure1A = Figure1A[["index","Product Name (place in the order of overall most common)", "Ing
#Figure1A = Figure1A[Figure1A['Ingredient 1'].notna()]
Figure1A
```

```
G = nx.Graph()
```

```
G.add_nodes_from(Figure1A['Name'], bipartite=1)
G.add_nodes_from(Figure1A['Ingredient1'], bipartite=0)

G.add_weighted_edges_from(
    [(row['Ingredient1'], row['Name'], 1) for idx, row in Figure1A.iterrows()],
    weight='index')

#G.add_weighted_edges_from(
```

```
#    [(row['Product Name (place in the order of overall most common)'], row['Ingredient 1'], 1
#    weight='weight')

print(G.edges(data=True))
```

```
[('Easy-Off Fume Free Oven Cleaner - Lemon Scent \n', 'Butane', {'index': 1}), ('Easy-Off
```

```
pos = {node:[0, i] for i,node in enumerate(Figure1A['Name'])}
pos.update({node:[1, i] for i,node in enumerate(Figure1A['Ingredient1'])})
for p in pos:  # raise text positions
    pos[p][1] +=0.05
nx.draw(G, pos, with_labels=True, node_size=0.1)
plt.rcParams["figure.figsize"] = (50,100)
plt.show()
```

```
nx.draw(G, pos, with_labels=False, node_size=0)
plt.rcParams["figure.figsize"] = (6,12)
plt.show()
```

Figure 1C Probability Distribution

The distribution of mixing size, capturing the number of cleaning product per mixing, across the 12 areas explored in our study.

```
import matplotlib.pyplot as plt
%matplotlib inline

import numpy as np
import pandas as pd
```

```
Figure1C = pd.read_excel('/content/Average number of product used for all tasks.xlsx')
Figure1C
```

	bathroom	kitchen
0	0.550095	0.639198
1	0.352552	0.294543
2	0.080340	0.065702
3	0.014650	0.000000
4	0.002363	0.000557

```
print (Figure1C.dtypes)
```

```
bathroom    float64
kitchen     float64
dtype: object
```

```
lines = Figure1C.plot.line()
plt.rcParams["figure.figsize"] = [14, 8]
plt.xticks([0.0, 1.0, 2.0, 3.0, 4.0],[1,2,3,4,5])    # changing x scale by own
plt.xlabel("Number of Mixing Cleaning Products per Area")
plt.ylabel("P(s)")
plt.show()
```

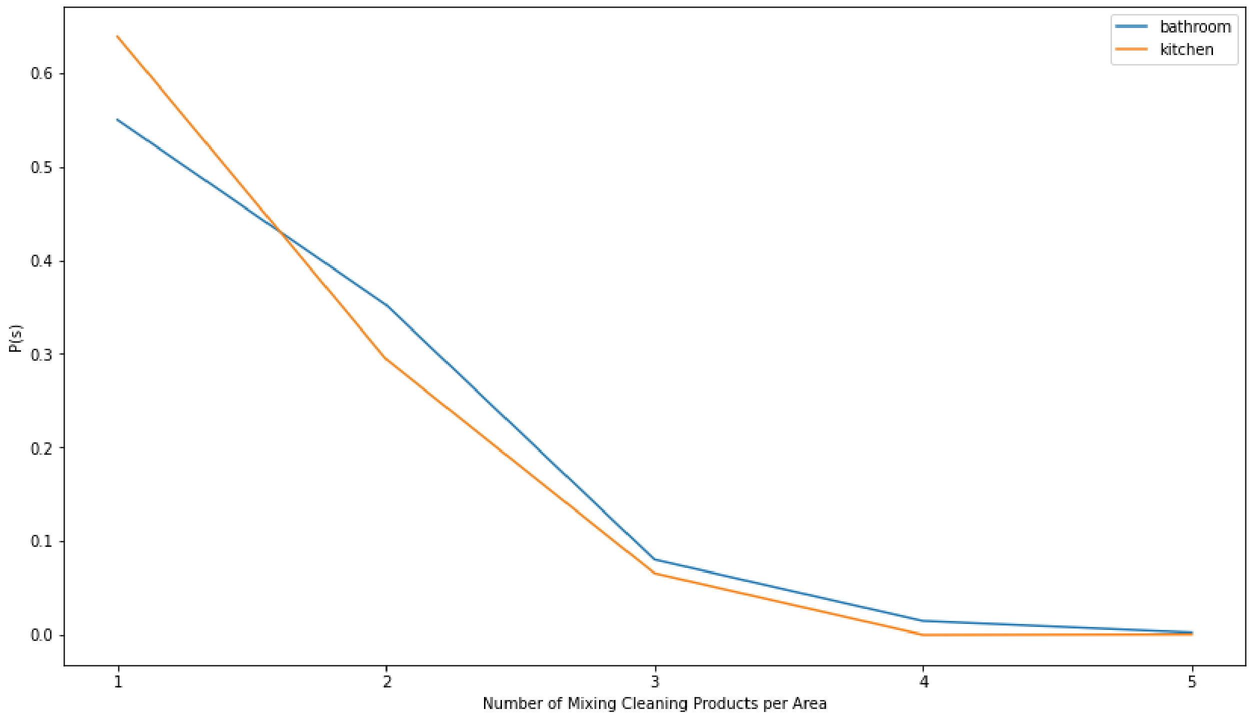


Figure 1D Frequency-Rank Plot

The frequency-rank plot of cleaning products across the 12 areas.

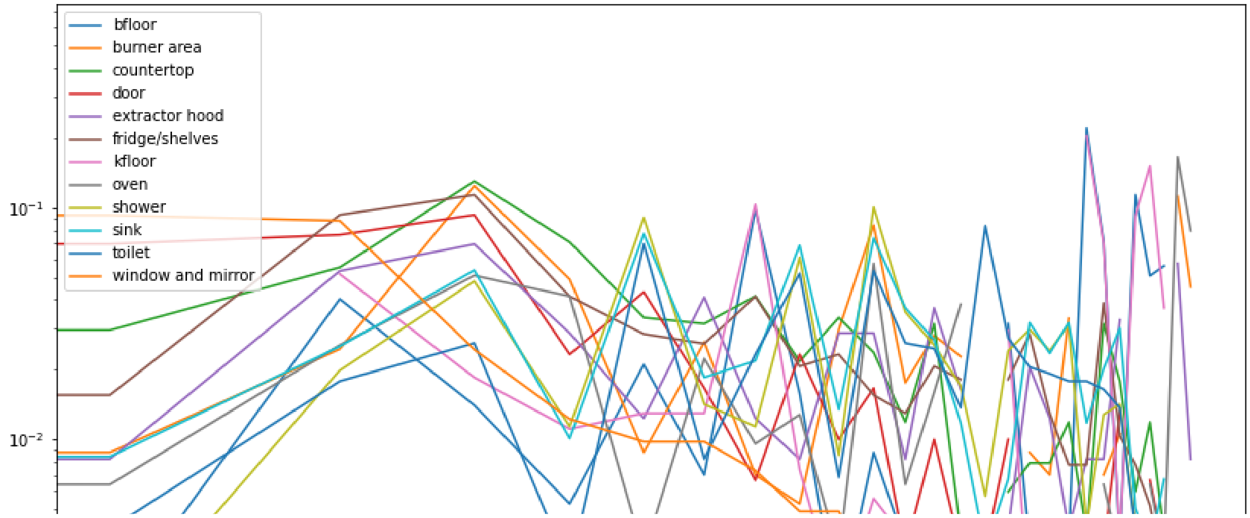
```
import matplotlib.pyplot as plt
%matplotlib inline

import numpy as np
import pandas as pd


Figure1D = pd.read_excel('/content/Top70PercentProductPercentage.xlsx')
Figure1D
```

	bfloor	burner area	countertop	door	extractor hood	fridge/shelves	kfloo
0	0.005254	0.029772	0.110454	0.299003	0.090164	0.131783	0.00000
1	0.014011	0.124343	0.130178	0.093023	0.069672	0.113695	0.01845
2	0.220665	0.000000	0.003945	0.000000	0.008197	0.007752	0.20479
3	0.008757	0.084063	0.023669	0.016611	0.028689	0.015504	0.00553
4	0.040280	0.024518	0.055227	0.076412	0.053279	0.093023	0.05166
5	0.021016	0.008757	0.033531	0.043189	0.012295	0.028424	0.01291
6	0.098074	0.007005	0.041420	0.006645	0.012295	0.041344	0.10332
7	0.015762	0.005254	0.021696	0.023256	0.008197	0.020672	0.00738
8	0.071804	0.007005	0.031558	0.003322	0.008197	0.038760	0.07011
9	0.005254	0.049037	0.071006	0.023256	0.028689	0.041344	0.01107
10	0.050788	0.003503	0.011834	0.006645	0.008197	0.005168	0.15129
11	0.113835	0.000000	0.005917	0.000000	0.000000	0.007752	0.09225
12	0.000000	0.112084	0.000000	0.000000	0.057377	0.000000	0.00000
13	0.001751	0.028021	0.031558	0.009967	0.036885	0.020672	0.00000
14	0.007005	0.026270	0.031558	0.016611	0.040984	0.025840	0.01291
15	0.001751	0.008757	0.029586	0.069767	0.008197	0.015504	0.00000
16	0.003503	0.017513	0.011834	0.003322	0.008197	0.012920	0.00369
17	0.003503	0.033275	0.011834	0.000000	0.004098	0.007752	0.00000
18	0.031524	0.000000	0.005917	0.009967	0.004098	0.018088	0.02952
19	0.003503	0.008757	0.007890	0.000000	0.020492	0.028424	0.00184
20	0.001751	0.029772	0.033531	0.009967	0.028689	0.023256	0.00184
21	0.003503	0.010508	0.017751	0.013289	0.032787	0.010336	0.00369
22	0.003503	0.022767	0.003945	0.003322	0.016393	0.018088	0.00000

```
lines = Figure1D.plot.line(loglog=True, legend='right')
```



References

Yong-Yeol Ahn, Sebastian E. Ahnert, James P. Bagrow & Albert-László Barabási. Flavor network and the principles of food pairing. Web page retrieved 06/08/2016 at http://www.nature.com/articles/srep00196.