Supplementary File S3: A graph-based approach to mapping human exposure-outcome associations for chemical contaminants

</center>

Taylor A. M. Wolffe[1,2], Paul Whaley[1,3], Crispin Halsall[1]

[1]Lancaster Environment Centre, Lancaster University, Lancaster, UK
[2]Yordas Group, Lancaster Environment Centre, Lancaster University, Lancaster, UK
[3]Evidence-Based Toxicology Collaboration, Johns Hopkins Bloomberg School of Public Health, Baltimore, MD 21205, USA

Querying the graph

First, all packages required for the processing of the raw data were imported. A connection with the Neo4j graph database was also established:

In [7]:

```
#Importing all required packages...
from py2neo import Graph, Node, Relationship
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns

#Connecting to the neo4j graph database...
graph = Graph("http://localhost:7474/db/data/", auth=('UserHere', 'YourPasswordHere'))
```

Included Publications and Number of Associations

In [15]:

```
#How many included Publications are there in the graph?
included_pubs_count = """
match (n:Publication)
return count(n)"""

results = graph.run(included_pubs_count)
qresults = results.to_data_frame()
qresults
```

Out[15]:

	count(n)
0	132

In [16]:

```
#How many associations are there in the graph?
included_assocs_count = """
match (n:Association)
return count(n)"""

results = graph.run(included_assocs_count)
qresults = results.to_data_frame()
qresults
```

Out[16]:

	count(n)
0	1656

In [17]:

```
#How many individual (single) chemical exposures are there in the graph?
chems_count ="""
match (n:SingleChemicalExposure)
return count(n)"""

results = graph.run(chems_count)
qresults = results.to_data_frame()
qresults
```

Out[17]:

	count(n)
0	326

In [20]:

```
#How many specific health outcomes are there?
outcomes_count ="""
match (n:HealthOutcome)
return count(n)"""

results = graph.run(outcomes_count)
qresults = results.to_data_frame()
qresults
```

Out[20]:

	count(n)
0	265

In [21]:

```
#What is the range and median for number of associations reported per publication?
range_median_assocs = '''
match (n:Publication)-[r:REPORTS]-(association)
return n.RefID as Reference, count(r) as NumberAssociations'''

results = graph.run(range_median_assocs)
qresults = results.to_data_frame()

print(qresults[qresults.NumberAssociations == qresults.NumberAssociations.min()])
print(qresults[qresults.NumberAssociations == qresults.NumberAssociations.max()])
print(qresults[qresults.NumberAssociations == qresults.NumberAssociations.median()])
```

```
                      Reference  NumberAssociations
18            Braun et al. 2006                   1
24            Min and Min 2013                    1
27        Froehlich et al. 2009                   1
37        Gallagher et al. 2011                   1
48            Braun et al. 2008                   1
56          Saraiva et al. 2007                   1
57   Gallagher and Meliker 2011                   1
58       Gallagher et al. 2010a                   1
67            Golub et al. 2010                   1
71               Ji et al. 2013                   1
88              Lee et al. 2012                   1
95          Teppala et al. 2012                   1
105      Lakind and Naiman 2011                   1
109   Bernard and McGeehin 2003                   1
110           Arora et al. 2009                   1
111                   Ford 2000                   1
120              Ng et al. 2013                   1
125      Gallagher et al. 2013b                   1
128         Mendola et al. 2013                   1
131        Bhandari et al. 2013                   1
            Reference  NumberAssociations
75  Mendy et al. 2012                 150
                      Reference  NumberAssociations
6       Navas-Acien et al. 2009                   4
8           Hoffman et al. 2010                   4
26        Steinmaus et al. 2009                   4
29   Gallagher and Meliker 2012                   4
30           Geiger et al. 2013                   4
38         Lanphear et al. 2000                   4
49             Moss et al. 1999                   4
50           Ballew et al. 1999                   4
62        Trasande et al. 2013b                   4
64      Fortenberry et al. 2012                   4
82              Lee et al. 2006                   4
85           JY Min et al. 2012                   4
87          Clayton et al. 2011                   4
90              Dye et al. 2002                   4
107                   Laks 2009                   4
117         Sudakin et al. 2013                   4
```

```
118           Menke et al. 2009                   4
119    Shargorodsky et al. 2011                   4
```

Exposure Queries

In [23]:

```
#Which chemical group has the largest number of associations across publications?
No_Assocs_ChemGroup = '''match (n:Association)-[r:ASSOCIATES]->(m)-[t:CODED_AS]->(p:SobusCode)
with n, p, m
return p.name as Name, count(n) as NoAssociations'''

results = graph.run(No_Assocs_ChemGroup)
Assocs_ChemGroup = results.to_data_frame()

Assocs_ChemGroup

#Visualised as Figure 3 in the manuscript which accompanies this Supplementary Information
```

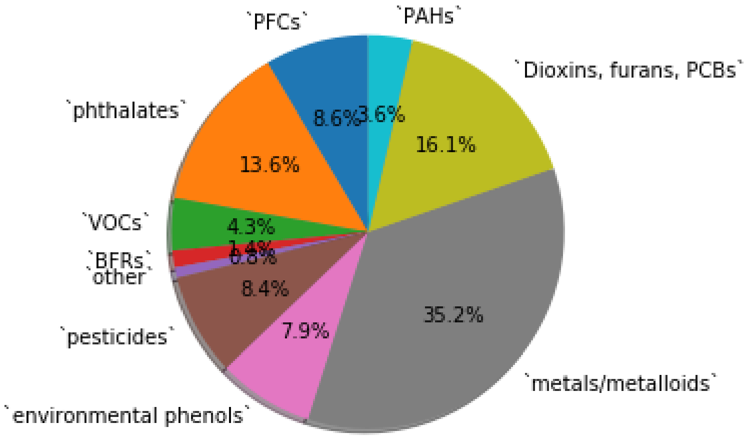
Out[23]:

	Name	NoAssociations
0	\`PFCs\`	142
1	\`phthalates\`	226
2	\`VOCs\`	71
3	\`BFRs\`	23
4	\`other\`	14
5	\`pesticides\`	139
6	\`environmental phenols\`	131
7	\`metals/metalloids\`	583
8	\`Dioxins, furans, PCBs\`	267
9	\`PAHs\`	60

In [25]:
```
labels = Assocs_ChemGroup.Name.tolist()
sizes = Assocs_ChemGroup.NoAssociations.tolist()

fig1, ax1 = plt.subplots()
ax1.pie(sizes,  labels=labels, autopct='%1.1f%%',
        shadow=True, startangle=90)
ax1.axis('equal')  # Equal aspect ratio ensures that pie is drawn as a circle.

plt.show()
```



In [26]:

```
#How are these associations distributed across publications?
pub_breakdown = '''match (a:Publication)-[s]->(n:Association)-[r:ASSOCIATES]->(m)-[t:CODED_AS]->(p:SobusCode)
with n, p, m, a
return p.name as Name, count(distinct a) as NoPubs'''


results = graph.run(pub_breakdown)
qresults = results.to_data_frame()
qresults
```

Out[26]:

	Name	NoPubs
0	`PFCs`	11
1	`phthalates`	12
2	`VOCs`	4
3	`BFRs`	2
4	`other`	5
5	`pesticides`	17
6	`environmental phenols`	18
7	`metals/metalloids`	84
8	`Dioxins, furans, PCBs`	9
9	`PAHs`	7

In [27]:

```
#How many chemicals make-up each exposure group?
NoChemsperGroup ='''match (n:Association)-[r:ASSOCIATES]->(m)-[t:CODED_AS]->(p:SobusCode)
with n, p, m
return p.name as Name, count(distinct m) as NoChems'''

results = graph.run(NoChemsperGroup)
qresults = results.to_data_frame()

qresults
```

Out[27]:

	Name	NoChems
0	`PFCs`	13
1	`phthalates`	38
2	`VOCs`	19
3	`BFRs`	7
4	`other`	7
5	`pesticides`	50
6	`environmental phenols`	17
7	`metals/metalloids`	47
8	`Dioxins, furans, PCBs`	110
9	`PAHs`	35

In [ ]:

```
#Which chemicals make up each exposure group and what is their frequency within the graph?
nochemstext = '''match (n:Association)-[r:ASSOCIATES]->(m)-[t:CODED_AS]->(p:SobusCode)
return p.name as Code, m.name as Chem, n.AssocID as Assocs'''

results = graph.run(nochemstext)
qresults = results.to_data_frame()

#Visualised as Supplementary File S4
```

In [33]:

```
#Whic chemical exposure groups are studied as mixtures most often?
ExpGroups = '''match (n:Association)-[r:ASSOCIATES]->(m:MixedChemicalExposure)-[t:CODED_AS]->(p:SobusCode)
with n, p, m
return p.name as Name, count(distinct m) as NoMixes'''

results = graph.run(ExpGroups)
qresults = results.to_data_frame()

qresults
```

Out[33]:

	Name	NoMixes
0	\`phthalates\`	10
1	\`pesticides\`	6
2	\`environmental phenols\`	2
3	\`metals/metalloids\`	4
4	\`Dioxins, furans, PCBs\`	12

In [36]:

```
#How many single chemicals were those mixtures typically associated with?
SingleChemsperGroupperExpGroup = '''match (n:Association)-[r:ASSOCIATES]->(m:MixedChemicalExposure)-[t:CODED_AS]->(p:SobusCode)
with n, p, m
match (m)-[s:COMPRISED_OF]-(l)
return p.name as Name, m.name as Mix, count(distinct l)'''

results = graph.run(SingleChemsperGroupperExpGroup)
qresults = results.to_data_frame()

qresults
```

Out[36]:

	Name	Mix	count(distinct l)
0	`Dioxins, furans, PCBs`	`Polychlorinated dibenzofurans (PCDFs), serum ...	3
1	`phthalates`	`Phthalates (high molecular weight (HMW)), uri...	6
2	`metals/metalloids`	`Lead & Cadmium, urine (Gollenberg et al. 2010)`	2
3	`metals/metalloids`	`Cadmium , urine AND blood (Ferraro et al. 2010)`	2
4	`phthalates`	`ΣDiethylhexyl phtalate (ΣDEHP), urine (Hoppin...	4
5	`Dioxins, furans, PCBs`	`Dioxin-like polychlorinated biphenyls (PCBs),...	8
6	`Dioxins, furans, PCBs`	`Non dioxin-like polychlorinated biphenyls (PC...	22
7	`phthalates`	`Phthalates, urine (Buttke et al. 2012)`	11
8	`metals/metalloids`	`Arsenic (total NOT Arsenobetaine), urine (Jon...	4
9	`Dioxins, furans, PCBs`	`Non dioxin-like polychlorinated biphenyls (PC...	5
10	`pesticides`	`Organochlorine pesticides, serum (Lee et al. ...	4
11	`phthalates`	`Phthalates (high molecular weight (HMW)), uri...	6
12	`phthalates`	`Phthalates (Low molecular weight (LMW)), urin...	3
13	`Dioxins, furans, PCBs`	`Non dioxin-like polychlorinated biphenyls (PC...	28
14	`Dioxins, furans, PCBs`	`Dioxin-like polychlorinated biphenyls (PCBs),...	4
15	`Dioxins, furans, PCBs`	`Polychlorinated dibenzodioxins (PCDDs), serum...	3
16	`phthalates`	`DBPCOM, urine (Ferguson et al. 2011)`	2
17	`pesticides`	`Organochlorine pesticides, serum (Lee et al. ...	7
18	`phthalates`	`Di-2-ethylhexylphthalate (DEHP) metabolites, ...	4
19	`Dioxins, furans, PCBs`	`Dioxin-like polychlorinated biphenyls (PCBs),...	9
20	`Dioxins, furans, PCBs`	`Polychlorinated dibenzofurans (PCDFs), serum ...	3
21	`environmental phenols`	`Parabens, urine (Buttke et al. 2012)`	2
22	`metals/metalloids`	`Arsenic (total), urine (Jones et al. 2011)`	5

	Name	Mix	count(distinct l)
23	`Dioxins, furans, PCBs`	`Polychlorinated dibenzodioxins (PCDDs), serum...	3
24	`phthalates`	`Phthalates (Low molecular weight (LMW)), urin...	3
25	`pesticides`	`Diethyl alkylphosphate (DEAP), urine (Bouchar...	3
26	`Dioxins, furans, PCBs`	`PCB-196 & PCB-203, serum (Cave et al. 2010)`	2
27	`Dioxins, furans, PCBs`	`PCB-138 & PCB-158, serum (Cave et al. 2010)`	2
28	`environmental phenols`	`Environmental phenols, urine (Buttke et al. 2...	2
29	`pesticides`	`Dichlorophenols, urine (Jerschow et al. 2012)`	2
30	`phthalates`	`Mono-butyl phthalates (MBP), urine (Stahlhut ...	2
31	`pesticides`	`Dimethyl alkylphosphate (DMAP), urine (Boucha...	3
32	`phthalates`	`Di-2-ethylhexylphthalate (DEHP) metabolites, ...	4
33	`pesticides`	`ΣTotal Dialkyl phosphate (DAP), urine (Boucha...	6

Health Outcome Queries

In [38]:

```
#What health outcome category was most often employed to categorize an association?
HealthOutcomeFrequency = '''match (n:HealthOutcomeCode)<-[r]-(m)<-[t]-(s:Association)
return n.name as HealthOutcome, count(distinct s) as NoAssocs'''

results = graph.run(HealthOutcomeFrequency)
qresults = results.to_data_frame()

#Visualised as Figure 4 in the manuscript which accompanies this supplementary information
qresults
```

```
Out[38]:
```

	HealthOutcome	NoAssocs
0	\`Bones and Joints\`	35
1	\`Blood\`	27
2	\`Teeth and Oral Health\`	10
3	\`Heart and circulatory\`	216
4	\`Kidneys\`	51
5	\`Reproductive System\`	163
6	\`Body Weight and Metabolism\`	681
7	\`Cognition and Mental Health\`	39
8	\`Other\`	12
9	\`Cancer\`	58
10	\`Mortality\`	81
11	\`Gene Expression\`	3
12	\`Audio-Visual System\`	16
13	\`Liver\`	136
14	\`Heart and Circulatory\`	17
15	\`Endocrine System\`	245
16	\`Lungs\`	93
17	\`Imunne System\`	84

In [ ]:

```
#Which specific outcomes make up each health outcome group and what is their frequency within the graph?
AllHealthOutcomes = '''match (n:Association)-[r:ASSOCIATES]->(m)-[t:CODED_AS]->(p:HealthOutcomeCode)
return p.name as Code, m.name as Outcome, n.AssocID as Assocs'''

results = graph.run(AllHealthOutcomes)
qresults = results.to_data_frame()

#visualised as Supplementary File S5
```

```
In [39]: #Which health outcome category was the most diverse?
         diversity_of_outcome_categories = '''match (n:HealthOutcomeCode)<-[r]-(m)<-[t]-(s:Association)
         return n.name as HealthOutcome, count(distinct m) as NoOutcomes'''

         results = graph.run(diversity_of_outcome_categories)
         qresults = results.to_data_frame()

         qresults
```

Out[39]:

	HealthOutcome	NoOutcomes
0	\`Bones and Joints\`	11
1	\`Blood\`	15
2	\`Teeth and Oral Health\`	10
3	\`Heart and circulatory\`	21
4	\`Kidneys\`	12
5	\`Reproductive System\`	19
6	\`Body Weight and Metabolism\`	61
7	\`Cognition and Mental Health\`	21
8	\`Other\`	5
9	\`Cancer\`	23
10	\`Mortality\`	33
11	\`Gene Expression\`	3
12	\`Audio-Visual System\`	2
13	\`Liver\`	15
14	\`Heart and Circulatory\`	9
15	\`Endocrine System\`	26
16	\`Lungs\`	16
17	\`Imunne System\`	23

In [40]:

```
#how are the health outcomes distributed across publications?
health_outcome_pub = '''match (a:Publication)-[s]->(n:Association)-[r:ASSOCIATES]->(m)-[t:CODED_AS]->(p:HealthOutcomeCode)
with n, p, m, a
return p.name as Name, count(distinct a) as NoPubs'''

results = graph.run(health_outcome_pub)
qresults = results.to_data_frame()

qresults
```

Out[40]:

	Name	NoPubs
0	\`Bones and Joints\`	8
1	\`Blood\`	11
2	\`Teeth and Oral Health\`	4
3	\`Heart and circulatory\`	26
4	\`Kidneys\`	10
5	\`Reproductive System\`	9
6	\`Body Weight and Metabolism\`	34
7	\`Cognition and Mental Health\`	13
8	\`Other\`	4
9	\`Cancer\`	14
10	\`Mortality\`	16
11	\`Gene Expression\`	2
12	\`Audio-Visual System\`	3
13	\`Liver\`	12
14	\`Heart and Circulatory\`	9
15	\`Endocrine System\`	16
16	\`Lungs\`	10
17	\`Imunne System\`	16

Association Queries

In [15]:

```
#Which exposure-outcome pairs (coded groups) were investigated most often?
exp_outcome_cat_frequency = '''match (n:HealthOutcomeCode)<-[r]-(m:HealthOutcome)<-[t]-(s:Association)-[q]-(l)-[w]->
(d:SobusCode)
return n.name as HealthOutcomeCat, d.name as ChemicalExpCat, count(distinct s) as NoAssocs'''

results = graph.run(exp_outcome_cat_frequency)
qresults = results.to_data_frame()

#Visualised as Figure 5a in the manuscript which accompanies this supplementary information
```

In [ ]:

```
#How are these associations broken down by publication?
exp_outcome_cat_pub_frequency = '''match (n:HealthOutcomeCode)<-[r]-(m:HealthOutcome)<-[t]-(s:Association)-[q]-(l)-[w]
->(d:SobusCode)
with n, m, s, d, l
match (a:Publication)-[o]->(s)
return n.name as HealthOutcomeCat, d.name as ChemicalExpCat, count(distinct a) as NoPubs'''

results = graph.run(exp_outcome_cat_pub_frequency)
qresults = results.to_data_frame()

#Visualised as Figure 5b in the manuscript which accompanies this supplementary information
```