Computing Features

```
// Find all C(ur)iaD pairs
MATCH (n0:Compound), (n3:Disease)
// Extract paths following the specified metapath
OPTIONAL MATCH paths = (n0: Compound)-[:UPREGULATES]-(n1)-[:INTERACTS]-
(n2)-[:ASSOCIATES]-(n3:Disease)
WITH
// reidentify the source and target nodes
n0 AS source,
n3 AS target,
paths,
// Extract the degrees along each path
[
size((n0)-[:UPREGULATES]-()),
size()-[:UPREGULATES]-(n1)),
size((n1)-[:INTERACTS]-()),
size()-[:INTERACTS]-(n2)),
size((n2)-[:ASSOCIATES]-()),
size()-[:ASSOCIATES]-(n3))
] AS degrees
RETURN
// Retrieve the name of the drug and disease
source.c_id AS compound_id,
source.name AS compound_name,
target.d_id AS disease_id,
target.name AS disease_name,
"CurGiGaD" AS metapath,
// Retrieve whether the drug treats the disease
size((source)-[:TREATS]-(target)) AS treatment,
// Compute the path count
count(paths) AS path_count,
// Compute the degree-weighted path count with w = 0.5
sum(reduce(pdp = 1.0, d in degreesl pdp * d ^ -0.5)) AS DWPC
// Sort the rows
ORDER BY DWPC DESC
```

```
// Find all C(dr)iaD pairs
MATCH (n0:Compound), (n3:Disease)
// Extract paths following the specified metapath
OPTIONAL MATCH paths = (n0: Compound)-[:DOWNREGULATES]-(n1)-
[:INTERACTS]-(n2)-[:ASSOCIATES]-(n3:Disease)
```

```
WITH
// reidentify the source and target nodes
n0 AS source,
n3 AS target,
paths,
// Extract the degrees along each path
[
size((n0)-[:DOWNREGULATES]-()),
size(()-[:DOWNREGULATES]-(n1)),
size((n1)-[:INTERACTS]-()),
size(()-[:INTERACTS]-(n2)),
size((n2)-[:ASSOCIATES]-()),
size(()-[:ASSOCIATES]-(n3))
] AS degrees
RETURN
// Retrieve the name of the drug and disease
source.c_id AS compound_id,
source.name AS compound_name,
target.d_id AS disease_id,
target.name AS disease_name,
"CdrGiGaD" AS metapath,
// Retrieve whether the drug treats the disease
size((source)-[:TREATS]-(target)) AS treatment,
// Compute the path count
count(paths) AS path_count,
// Compute the degree-weighted path count with w = 0.5
sum(reduce(pdp = 1.0, d in degreesl pdp * d ^ -0.5)) AS DWPC
// Sort the rows
ORDER BY DWPC DESC
```

```
// Find all CsCtD pairs
MATCH (n0:Compound), (n3:Disease)
// Extract paths following the specified metapath
OPTIONAL MATCH paths = (n0)-[:CAUSES]-(n1:SideEffect)-[:CAUSES]-(n2:
Compound)-[:TREATS]-(n3)
// Specify the join index to reach lightspeed
USING JOIN ON n1
// Exclude paths with duplicate nodes
WHERE n0 <> n2
WITH
  // reidentify the source and target nodes
  n0 AS source,
  n3 AS target,
  paths,
```

```
  // Extract the degrees along each path
  [
    size((n0)-[:CAUSES]-()),
    size(()-[:CAUSES]-(n1)),
    size((n1)-[:CAUSES]-()),
    size(()-[:CAUSES]-(n2)),
    size((n2)-[:TREATS]-()),
    size(()-[:TREATS]-(n3))
  ] AS degrees
RETURN
  source.c_id AS compound_id,
  source.name AS compound_name,
  target.d_id AS disease_id,
  target.name AS disease_name,
  "CsCtD" AS metapath,
  // Retrieve whether the drug treats the disease
  size((source)-[:TREATS]-(target)) AS treatment,
  // Compute the path count
  count(paths) AS path_count,
  // Compute the degree-weighted path count with w = 0.5
  sum(reduce(pdp = 1.0, d in degreesl pdp * d ^ -0.5)) AS DWPC
// Sort the rows
ORDER BY DWPC DESC
```

// Find all CtGaD pairs

```
MATCH (n0:Compound), (n2:Disease)
// Extract paths where the drug targets a gene associated with the disease
OPTIONAL MATCH paths = (n0:Compound)-[:TARGETS]-(n1:Gene)-[:ASSOCIATES]-
(n2:Disease)
WITH
// reidentify the source and target nodes
n0 AS source,
n2 AS target,
paths,
[
size((n0)-[:TARGETS]-()),
size(()-[:TARGETS]-(n1)),
size((n1)-[:ASSOCIATES]-()),
size(()-[:ASSOCIATES]-(n2))
] AS degrees
RETURN
// Retrieve the name of the drug and disease
source.c_id AS compound_id,
source.name AS compound_name,
```

```
target.d_id AS disease_id,
target.name AS disease_name,
"CtGaD" AS metapath,

// Retrieve whether the drug treats the disease
size((source)-[:TREATS]-(target)) AS treatment,
// Count the number of paths between the drug and disease
count(paths) AS path_count,
// Compute the degree-weighted path count with w = 0.5
sum(reduce(pdp = 1.0, d in degreesl pdp * d ^ -0.5)) AS DWPC
// Sort the rows
ORDER BY path_count DESC, treatment DESC
```
