

## Neo4J-commands:

### Data bulk insertion

#### **i1: Loading the CSV files (genes, enhancers, TFPairs)**

```
LOAD CSV FROM "file:///genenames.csv" AS line
CREATE (:Gene { genename: toString(line[0])});
```

#### **c1: Assigning a uniqueness constraints to nodes**

```
CREATE CONSTRAINT ON (g:Gene) ASSERT g.genename IS UNIQUE;
```

#### **i2: Loading the CSV files (genes, enhancers, TFPairs)**

```
LOAD CSV FROM "file:///enhids.csv" AS line
CREATE (:Enhancer { enhancerID: toString(line[0])});
```

#### **c2: Assigning a uniqueness constraints to nodes**

```
CREATE CONSTRAINT ON (e:Enhancer) ASSERT e.enhancerID IS UNIQUE;
```

#### **i3: Loading the CSV files (genes, enhancers, TFPairs)**

```
LOAD CSV FROM "file:///tfpairs.csv" AS line
CREATE (p:TFPair {name:toString(line[4]),
pwm1:toString(line[0]),
pwm2:toString(line[1]),
knownCompelPair:toBoolean(line[2]),
knownBioGridPair:toBoolean(line[3])})
RETURN p.name;
```

#### **c3: Assigning a uniqueness constraints to nodes**

```
CREATE CONSTRAINT ON (t:TFPair) ASSERT t.name IS UNIQUE;
```

#### **i4: Loading relationship data from CSV files (EPI and binds)**

```
LOAD CSV FROM "file:///enh2promo.csv" AS line
MATCH (e:Enhancer {enhancerID:toString(line[0])})
MATCH (g:Gene {genename:toString(line[1])})
MERGE (e)-[r:EPI]->(g) RETURN e.enhancerID, g.genename;
```

#### **i5: Loading relationship data from CSV files (EPI and binds)**

```
LOAD CSV FROM "file:///tf2gene.csv" AS line
MATCH (p:TFPair {name:toString(line[5])})
MATCH (g:Gene {genename:toString(line[4])})
CREATE (p)-[r:binds {zscore:toFloat(line[2])}]->(g)
RETURN p.name,g.genename,r.zscore;
```

#### **i6: Loading relationship data from CSV files (EPI and binds)**

```
LOAD CSV FROM "file:///tf2enh.csv" AS line
MATCH (p:TFPair {name:toString(line[4])})
MATCH (e:Enhancer {enhancerID:toString(line[3])})
CREATE (p)-[r:binds {zscore:toFloat(line[2])}]->(e)
RETURN p.name,e.enhancerID,r.zscore;
```

### **Path queries**

**Q1a: For a given genename, find all enhancers interacting with that gene.**

```
MATCH (g:Gene)-[:EPI]-(e:Enhancer)
WHERE g.genename="SLC25A45" RETURN g,e
```

**Q1b: For a genename set, find all enhancers interacting with the genes.**

```
MATCH (g:Gene)-[:EPI]-(e:Enhancer)
WHERE g.genename IN ["IER5L","SLC25A45"] RETURN g,e
```

**Q2a: For a given genename, find all TFPairs bound to that gene.**

```
MATCH (g:Gene)-[:binds]-(t:TFPair)
WHERE g.genename="SLC25A45" RETURN g,t
```

**Q2b: Restrict to the known TFPairs with AND operator.**

```
MATCH (g:Gene)-[:binds]-(t:TFPair)
WHERE g.genename="SLC25A45"
AND t.knownCompelPair=TRUE RETURN g,t
```

**Q2c: Restrict to the known TFPairs with AND and OR operator**

```
MATCH (g:Gene)-[:binds]-(t:TFPair)
WHERE g.genename="SLC25A45"
AND (t.knownCompelPair=TRUE OR t.knownBioGridPair=TRUE)
RETURN g,t
```

**Q2d: Find the TFPairs of an enhancer that interact with a certain gene.**

```
MATCH p=(g:Gene)-[:EPI]-(e:Enhancer)-[:binds]-(t:TFPair)
WHERE g.genename="BATF2" RETURN p
```

**Q2e: Restrict to z-score larger than 4.**

```
MATCH p=(g:Gene)-[:EPI]-(e:Enhancer)-[:binds]-(t:TFPair)
WHERE g.genename="BATF2" AND r.zscore>4 RETURN p
```

**Q3a: For all genenames find all other genenames that are bound by at least one common TFPair.**

```
MATCH p=(g:Gene)-[:binds]-(t:TFPair)-[:binds]-(g2:Gene)
RETURN p
```

**Q3b: For a specific gene find all other genenames that are bound by at least one common TFPair.**

```
MATCH p=(g:Gene)-[:binds]-(t:TFPair)-[:binds]-(g2:Gene)
WHERE g.genename="MYL12B" RETURN p
```

**Q3c: For a specific enhancerID find all other enhancerIDs that are bound by at least one common TFPair.**

```
MATCH p=(e:Enhancer)-[:binds]-(t:TFPair)-[:binds]-(e2:Enhancer)
WHERE e.enhancerID="chr11:65239550-65239870-enhancer"
RETURN p
```

**Q3d: For a specific enhancerID find genenames that are bound by at least one common TFPair.**

```
MATCH p=(e:Enhancer)-[:binds]-(t:TFPair)-[:binds]-(g:Gene)
WHERE e.enhancerID="chr11:65239550-65239870-enhancer"
```

RETURN p

**Q4a: For a given enhancer ID (or a prefix of the ID), find all the TFPairs bound to the enhancer.**

MATCH (e:Enhancer)-[:binds]-(t:TFPair)

WHERE e.enhancerID STARTS WITH "chr1:" RETURN e,t

**Q5a: For a given enhancerID, find all genes interacting with the enhancer.**

MATCH (e:Enhancer)-[:EPI]-(g:Gene)

WHERE e.enhancerID STARTS WITH "chr11:" RETURN e,g

**Q6a: For a given genename, find all TFPairs bound to the gene.**

MATCH p=(g:Gene)-[r:binds]-(t:TFPair)

WHERE g.genename="SLC25A45" RETURN p

**Q6b: For a given genename, find all TFPairs bound to the gene restricting to those bindings with a high zscore.**

MATCH p=(g:Gene)-[r:binds]-(t:TFPair)

WHERE g.genename="SLC25A45" AND r.zscore>4 RETURN p

**Q7a: For a given TF find all TFPairs that contain the TF.**

MATCH (t:TFPair)

WHERE t.pwm1="V\$ELK1\_02" OR t.pwm2="V\$ELK1\_02"

RETURN t

**Q7b: For a given TF find the names of the two transcription factors in the TFPairs that contain the transcription factor.**

MATCH (t:TFPair)

WHERE t.pwm1="V\$ELK1\_02" OR t.pwm2="V\$ELK1\_02"

RETURN t.pwm1,t.pwm2

### Statistical queries

**G1a: Count the total number of TFPairs that one enhancer has in common with any other.**

MATCH p=(e:Enhancer)-[:binds]-(t:TFPair)-[:binds]-(e2:Enhancer)

RETURN e.enhancerID, COUNT(t)

**G1b: Count the TFPairs that two specific enhancers have in common**

MATCH p=(e:Enhancer)-[:binds]-(t:TFPair)-[:binds]-(e2:Enhancer)

RETURN e.enhancerID, COUNT(t), e2.enhancerID