Create ecocyc-lei

July 7, 2022

1 LL-3821 Tracing for Lei - Generate graphdb ecocyc-lei

https://sbrgsoftware.atlassian.net/browse/LL-3821

The database ecocyc-lei was generated by removing and adding nodes to ecocyc-mod2. Results will not exactly match ecocyc-25.5-gds.

First, create a database named 'ecocyc-lei' that is the same as ecocyc-mod2

[17]:
```
import pandas as pd
from py2neo import Graph
pd.set_option('display.max_colwidth', 170)
```

[18]:
```
graph = Graph("bolt://localhost:7687", auth=("neo4j", "rcai"),␣
↪name="ecocyc-lei")
graph_ll = Graph("bolt://localhost:7687", auth=("neo4j", "rcai"),␣
↪name="lifelike-stg")
graph_humancyc = Graph("bolt://localhost:7687", auth=("neo4j", "rcai"),␣
↪name="humancyc")
```

1.0.1 Knockout genes: tnaA, yddG, IdhA, CytR

Knockout the list of genes, and their encoded proteins and catalyzed reactions

1.0.2 Get biocyc ids for the genes, proteins and reactions

Get all nodes downstream of the given genes (proteins, reactions)

[6]:
```
def get_biocyc_ids(geneName):
    query = """
    match p=(n:Gene {name: $geneName})-[:ENCODES]-()-[:COMPONENT_OF*0..]->()-[:
↪CATALYZES|CONSUMED_BY]-()
    with nodes(p) as nodes unwind nodes as n
    return distinct n.biocyc_id, n.name, n.entityType
    """
    df = graph.run(query, geneName=geneName).to_data_frame()
    print(df)
    ids = [id for id in df['n.biocyc_id']]
    print(ids)
```

```
    return ids
```

[18]:
```
df1 = get_biocyc_ids('tnaA')
```

```
          n.biocyc_id                    n.name n.entityType
0             EG11005                      tnaA         Gene
1  TRYPTOPHAN-MONOMER              tryptophanase      Protein
2     TRYPTOPHAN-CPLX  complex of tryptophanase      Protein
3      TRYPTOPHAN-RXN             TRYPTOPHAN-RXN     Reaction
4      LCYSDESULF-RXN             LCYSDESULF-RXN     Reaction
['EG11005', 'TRYPTOPHAN-MONOMER', 'TRYPTOPHAN-CPLX', 'TRYPTOPHAN-RXN',
'LCYSDESULF-RXN']
```

Remove [‘EG11005’, ‘TRYPTOPHAN-MONOMER’, ‘TRYPTOPHAN-CPLX’, ‘TRYPTOPHAN-RXN’] Run the following cypher query

```
match(n:db_EcoCyc) where n.biocyc_id in  ['EG11005', 'TRYPTOPHAN-MONOMER', 'TRYPTOPHAN-CPLX',
```

[19]:
```
df = get_biocyc_ids('yddG')
```

```
       n.biocyc_id                     n.name n.entityType
0          EG12713                       yddG         Gene
1  EG12713-MONOMER  amino acid exporter YddG      Protein
2   TRANS-RXN0-265          amino acid export     Reaction
['EG12713', 'EG12713-MONOMER', 'TRANS-RXN0-265']
```

Remove [‘EG12713’, ‘EG12713-MONOMER’, ‘TRANS-RXN0-265’]

```
match(n:db_EcoCyc) where n.biocyc_id in  ['EG12713', 'EG12713-MONOMER', 'TRANS-RXN0-265'] deta
```

[20]:
```
df = get_biocyc_ids('ldhA')
```

```
                n.biocyc_id                     n.name n.entityType
0                      G592                       ldhA         Gene
1  DLACTDEHYDROGNAD-MONOMER  D-lactate dehydrogenase      Protein
2                CPLX0-8158  D-lactate dehydrogenase      Protein
3      DLACTDEHYDROGNAD-RXN     DLACTDEHYDROGNAD-RXN     Reaction
['G592', 'DLACTDEHYDROGNAD-MONOMER', 'CPLX0-8158', 'DLACTDEHYDROGNAD-RXN']
```

Remove the following biocyc id (reaction is reversible, has two entries) [‘G592’, ‘DLACTDEHYDROGNAD-MONOMER’, ‘CPLX0-8158’, ‘DLACTDEHYDROGNAD-RXN’]

```
match(n:db_EcoCyc) where n.biocyc_id in
['G592', 'DLACTDEHYDROGNAD-MONOMER', 'CPLX0-8158', 'DLACTDEHYDROGNAD-RXN']
detach delete n;
```

[15]:
```
df = get_biocyc_ids('cytR')
```

```
  n.biocyc_id                                        n.name n.entityType
0     EG10200                                         cytR         None
```

```
1     PD04028  DNA-binding transcriptional repressor CytR         None
2  CPLX0-7740  DNA-binding transcriptional repressor CytR         None
3   RXN0-6409                                          None         None
4  CPLX0-8051                                CytR-cytidine         None
['EG10200', 'PD04028', 'CPLX0-7740', 'RXN0-6409', 'CPLX0-8051']
```

Remove [‘EG10200’, ‘PD04028’, ‘CPLX0-7740’, ‘RXN0-6409’, ‘CPLX0-8051’]

```
match(n:db_EcoCyc) where n.biocyc_id in
['EG10200', 'PD04028', 'CPLX0-7740', 'RXN0-6409', 'CPLX0-8051']
detach delete n;
```

1.1 Knock-Ins: DDC, AANAT, ASMT and phhB

[25]:
```
def get_reaction_nodes(reaction_id):
    """
    Given the reaction id (humancyc reaction), find all the associated␣
 ↪compounds.
    Query used lifelike database
    """
    query = """
    match p=(r:db_BioCyc {biocyc_id:$reaction_id})-[:CONSUMED_BY|PRODUCES]-()
    unwind nodes(p) as n
    return distinct n.biocyc_id,n.name, labels(n)
    """
    df = graph_ll.run(query, reaction_id=reaction_id).to_data_frame()
    return df
```

1. Reaction: RXN3DJ-170 catalyzed by DDC Both serotonin and 5-hydroxyl-L-tryptophan are not ecocyc compound

[30]:
```
df = get_reaction_nodes('RXN3DJ-170')
print(len(df))
df
```

```
5
```

[30]:

	n.biocyc_id	n.name	\
0	RXN3DJ-170	None	
1	PROTON	H+	
2	5-HYDROXY-TRYPTOPHAN	5-hydroxy-L-tryptophan	
3	CARBON-DIOXIDE	CO2	
4	SEROTONIN	serotonin	

	labels(n)
0	[db_BioCyc, db_HumanCyc, db_PseudomonasCyc, Reaction]
1	[db_BioCyc, db_EcoCyc, Compound, db_HumanCyc, db_YeastCyc, db_PseudomonasCyc]
2	[db_BioCyc, Compound, db_HumanCyc, db_PseudomonasCyc]

```
3  [db_BioCyc, db_EcoCyc, Compound, db_HumanCyc, db_YeastCyc, db_PseudomonasCyc]
4                          [db_BioCyc, Compound, db_HumanCyc, db_PseudomonasCyc]
```

2. Reaction AROMATIC-L-AMINO-ACID-DECARBOXYLASE-RXN tryptamine is not EcoCyc compound

[31]:
```
get_reaction_nodes("AROMATIC-L-AMINO-ACID-DECARBOXYLASE-RXN")
```

[31]:
```
                              n.biocyc_id        n.name  \
0  AROMATIC-L-AMINO-ACID-DECARBOXYLASE-RXN          None
1                                     TRP  L-tryptophan
2                                  PROTON            H+
3                               TRYPTAMINE    tryptamine
4                           CARBON-DIOXIDE           CO2

                                                                  labels(n)
0                          [db_BioCyc, db_HumanCyc, db_PseudomonasCyc, Reaction]
1  [db_BioCyc, db_EcoCyc, Compound, db_HumanCyc, db_YeastCyc, db_PseudomonasCyc]
2  [db_BioCyc, db_EcoCyc, Compound, db_HumanCyc, db_YeastCyc, db_PseudomonasCyc]
3                          [db_BioCyc, Compound, db_HumanCyc, db_PseudomonasCyc]
4  [db_BioCyc, db_EcoCyc, Compound, db_HumanCyc, db_YeastCyc, db_PseudomonasCyc]
```

1.1.1 Based on the spread sheet, create the following reactions:

- PHHB_RXN: tryptonphan -> 5-hydroxytryptophan
- DDC_RXN: 5-hydroxytryptophan -> 5-hydroxytryptamine + CO2
- AANAT_RXN: 5-hydroxytryptamine + acetyl-CoA -> acetyl-5-hydroxytryptamine + CoA
- ASMT_RXN: acetyl-5-hydroxytryptamine + SAM -> melatonin + SAH

[32]:
```
# get all compounds from BioCyc:
query = """
match(n:Synonym) where n.name in
['tryptophan', 'L-5-hydroxytryptophan', '5-hydroxytryptamine',␣
 ↪'N-acetyl-5-hydroxytryptamine', 'melatonin', 'SAM', 'SAH', 'CO2', 'CoA',␣
 ↪'acetyl-CoA']
with n optional match(n)-[:HAS_SYNONYM]-(b:Compound)
return n.name, b.biocyc_id, b.name, labels(b)
"""
df = graph_ll.run(query).to_data_frame()
```

[33]:
```
df
```

[33]:
```
                        n.name                    b.biocyc_id  \
0                    tryptophan                            TRP
1         L-5-hydroxytryptophan          5-HYDROXY-TRYPTOPHAN
```

```
2           5-hydroxytryptamine                       SEROTONIN
3  N-acetyl-5-hydroxytryptamine              N-ACETYL-SEROTONIN
4                     melatonin  N-ACETYL-5-METHOXY-TRYPTAMINE
5                           SAM            S-ADENOSYLMETHIONINE
6                           SAH               ADENOSYL-HOMO-CYS
7                           CO2                  CARBON-DIOXIDE
8                           CoA                           CO-A
9                     acetyl-CoA                      ACETYL-COA

                      b.name  \
0               L-tryptophan
1     5-hydroxy-L-tryptophan
2                  serotonin
3         N-acetyl-serotonin
4                  melatonin
5    S-adenosyl-L-methionine
6  S-adenosyl-L-homocysteine
7                        CO2
8                 coenzyme A
9                 acetyl-CoA

    labels(b)
0               [db_BioCyc, db_EcoCyc, Compound, db_HumanCyc, db_YeastCyc,
db_PseudomonasCyc]
1                                        [db_BioCyc, Compound, db_HumanCyc,
db_PseudomonasCyc]
2                                        [db_BioCyc, Compound, db_HumanCyc,
db_PseudomonasCyc]
3                                                             [db_BioCyc, Compound,
db_HumanCyc]
4                                                             [db_BioCyc, Compound,
db_HumanCyc]
5  [db_BioCyc, db_EcoCyc, Compound, db_HumanCyc, db_YeastCyc, db_PseudomonasCyc,
BioCycClass]
6               [db_BioCyc, db_EcoCyc, Compound, db_HumanCyc, db_YeastCyc,
db_PseudomonasCyc]
7               [db_BioCyc, db_EcoCyc, Compound, db_HumanCyc, db_YeastCyc,
db_PseudomonasCyc]
8               [db_BioCyc, db_EcoCyc, Compound, db_HumanCyc, db_YeastCyc,
db_PseudomonasCyc]
9               [db_BioCyc, db_EcoCyc, Compound, db_HumanCyc, db_YeastCyc,
db_PseudomonasCyc]
```

1.1.2 Create pathway - PMY-6030

[48]:
```
query = """
match (n:Pathway:db_BioCyc) where n.biocyc_id = 'PWY-6030'
return n.biocyc_id as biocyc_id, n.name as name
"""
pathway = graph_humancyc.run(query).data()[0]
pathway
```

[48]:
```
{'biocyc_id': 'PWY-6030', 'name': 'serotonin and melatonin biosynthesis'}
```

[49]:
```
query = """
merge (n:Pathway:db_BioCyc {biocyc_id:$biocyc_id})
set n.eid = $biocyc_id, n.name = $name, n.dispayName=$name, n.
 ↪entityType='Pathway', n.comment='knocked in'
"""
graph.run(query, biocyc_id=pathway['biocyc_id'], name=pathway['name'])
```

[49]:
```
(No data)
```

1.1.3 Create reactions

[47]:
```
query = """
match (n:Pathway:db_BioCyc) where n.biocyc_id = 'PWY-6030' with n
match (n)-[:IN_PATHWAY]-(r:Reaction)-[]-(:EnzReaction)-[:CATALYZES]-()<-[:
 ↪COMPONENT_OF*0..]-()-[:ENCODES]-(g)
return [x in collect(g) | x.name] as gene, r.biocyc_id as biocyc_id, r.
 ↪ec_number as ec_number,
r.direction as direction, replace(r.description, "==", "=") as detail
"""
reactions = graph_humancyc.run(query).to_data_frame()
reactions.to_excel("/Users/rcai/data/notebook/lei/humancyc_melatonin_rxn.xlsx",␣
 ↪index=False)
```

[54]:
```
query = """
match (p:Pathway {biocyc_id: $pathway_id})
with p match (p)-[:IN_PATHWAY]-(r:Reaction)-[:COMSUMED_BY|PRODUCES]-(c:Compound)
return c.biocyc_id as biocyc_id, c.name as name
"""
chems = graph_humancyc.run(query, pathway_id=pathway['biocyc_id']).data()
```

[79]:
```
def add_compounds(pathway_id):
    query = """
    match (p:Pathway {biocyc_id: $pathway_id})
    with p match (p)-[:IN_PATHWAY]-(r:Reaction)-[:CONSUMED_BY|PRODUCES]-(c:
 ↪Compound)
    return c.biocyc_id as biocyc_id, c.name as name
```

```
    """
    chems = graph_humancyc.run(query, pathway_id=pathway_id).data()

    query2 = """
    merge (c:Compound {biocyc_id:$biocyc_id}) set c:db_BioCyc, c.name=$name,
    c.displayName=$name, c.eid=$biocyc_id
    """
    for chem in chems:
        print("add", chem)
        graph.run(query2, biocyc_id=chem['biocyc_id'], name=chem['name'])
```

[80]:
```
add_compounds(pathway['biocyc_id'])
```

```
add {'biocyc_id': 'CARBON-DIOXIDE', 'name': 'CO2'}
add {'biocyc_id': 'SEROTONIN', 'name': 'serotonin'}
add {'biocyc_id': '5-HYDROXY-TRYPTOPHAN', 'name': '5-hydroxy-L-tryptophan'}
add {'biocyc_id': 'PROTON', 'name': 'H+'}
add {'biocyc_id': 'N-ACETYL-SEROTONIN', 'name': 'N-acetyl-serotonin'}
add {'biocyc_id': 'PROTON', 'name': 'H+'}
add {'biocyc_id': 'CO-A', 'name': 'coenzyme A'}
add {'biocyc_id': 'ACETYL-COA', 'name': 'acetyl-CoA'}
add {'biocyc_id': 'SEROTONIN', 'name': 'serotonin'}
add {'biocyc_id': 'PROTON', 'name': 'H+'}
add {'biocyc_id': 'ADENOSYL-HOMO-CYS', 'name': 'S-adenosyl-L-homocysteine'}
add {'biocyc_id': 'N-ACETYL-5-METHOXY-TRYPTAMINE', 'name': 'melatonin'}
add {'biocyc_id': 'N-ACETYL-SEROTONIN', 'name': 'N-acetyl-serotonin'}
add {'biocyc_id': '5-HYDROXY-TRYPTOPHAN', 'name': '5-hydroxy-L-tryptophan'}
add {'biocyc_id': 'CPD-5881', 'name': '(6R)-4a-hydroxy-tetrahydrobiopterin'}
add {'biocyc_id': 'TRP', 'name': 'L-tryptophan'}
add {'biocyc_id': 'OXYGEN-MOLECULE', 'name': 'oxygen'}
add {'biocyc_id': 'CPD-14053', 'name':
'(6R)-L-erythro-5,6,7,8-tetrahydrobiopterin'}
```

[83]:
```
def add_reactions(pathway_id):
    query = """
    match (p:Pathway {biocyc_id: $pathway_id})-[:IN_PATHWAY]-(r:Reaction)
    return r.biocyc_id as biocyc_id, r.ec_number as ec_number,
    replace(r.description, '==', '=') as detail
    """
    reactions = graph_humancyc.run(query, pathway_id=pathway_id).data()
    print('reactions:', len(reactions))
    query_rxn = """
    merge(n:Reaction {biocyc_id:$biocyc_id}) set n:db_BioCyc, n.eid =␣
↪$biocyc_id,
    n.ec_number=$ec_number, n.displayName=$ec_number, n.detail=$detail, n.
↪comment='knocked in'
    """
```

```
    query_pwy = """
    match(n:Reaction {biocyc_id:$biocyc_id}), (p:Pathway {biocyc_id:
↪$pathway_id})
    merge (n)-[:IN_PATHWAY]->(p)
    """
    for r in reactions:
        print(r['biocyc_id'])
        graph.run(query_rxn, biocyc_id=r['biocyc_id'],␣
↪ec_number=r['ec_number'], detail=r['detail'])
        graph.run(query_pwy, biocyc_id=r['biocyc_id'], pathway_id=pathway_id)

    # add input
    query = """
    match (p:Pathway {biocyc_id: $pathway_id})-[:IN_PATHWAY]-(r:Reaction)
    with r match (r)-[:CONSUMED_BY]-(c)
    return r.biocyc_id as reaction_id, c.biocyc_id as chem_id
    """
    inputs = graph_humancyc.run(query, pathway_id=pathway_id).data()
    print("inputs", len(inputs))

    query_input = """
    match(r:Reaction {biocyc_id: $reaction_id}), (c:db_BioCyc {biocyc_id:␣
↪$chem_id})
    merge (c)-[:CONSUMED_BY]->(r)
    """
    for inchem in inputs:
        print(inchem)
        graph.run(query_input, reaction_id=inchem['reaction_id'],␣
↪chem_id=inchem['chem_id'])

    # add output
    query = """
    match (p:Pathway {biocyc_id: $pathway_id})-[:IN_PATHWAY]-(r:Reaction)
    with r match (r)-[:PRODUCES]-(c)
    return r.biocyc_id as reaction_id, c.biocyc_id as chem_id
    """
    outputs = graph_humancyc.run(query, pathway_id=pathway_id).data()
    print("outputs", len(outputs))

    query_output = """
    match(r:Reaction {biocyc_id: $reaction_id}), (c:db_BioCyc {biocyc_id:␣
↪$chem_id})
    merge (r)-[:PRODUCES]->(c)
    """
    for outchem in outputs:
        print(outchem)
```

```
        graph.run(query_output, reaction_id=outchem['reaction_id'],␣
↪chem_id=outchem['chem_id'])
```

[84]:
```
add_reactions(pathway['biocyc_id'])
```

```
reactions: 4
RXN3DJ-170
RXN3DJ-35528
ACETYLSEROTONIN-O-METHYLTRANSFERASE-RXN
TRYPTOPHAN-5-MONOOXYGENASE-RXN
inputs 9
{'reaction_id': 'RXN3DJ-170', 'chem_id': '5-HYDROXY-TRYPTOPHAN'}
{'reaction_id': 'RXN3DJ-170', 'chem_id': 'PROTON'}
{'reaction_id': 'RXN3DJ-35528', 'chem_id': 'ACETYL-COA'}
{'reaction_id': 'RXN3DJ-35528', 'chem_id': 'SEROTONIN'}
{'reaction_id': 'ACETYLSEROTONIN-O-METHYLTRANSFERASE-RXN', 'chem_id':
'S-ADENOSYLMETHIONINE'}
{'reaction_id': 'ACETYLSEROTONIN-O-METHYLTRANSFERASE-RXN', 'chem_id': 'N-ACETYL-
SEROTONIN'}
{'reaction_id': 'TRYPTOPHAN-5-MONOOXYGENASE-RXN', 'chem_id': 'TRP'}
{'reaction_id': 'TRYPTOPHAN-5-MONOOXYGENASE-RXN', 'chem_id': 'OXYGEN-MOLECULE'}
{'reaction_id': 'TRYPTOPHAN-5-MONOOXYGENASE-RXN', 'chem_id': 'CPD-14053'}
outputs 10
{'reaction_id': 'RXN3DJ-170', 'chem_id': 'CARBON-DIOXIDE'}
{'reaction_id': 'RXN3DJ-170', 'chem_id': 'SEROTONIN'}
{'reaction_id': 'RXN3DJ-35528', 'chem_id': 'N-ACETYL-SEROTONIN'}
{'reaction_id': 'RXN3DJ-35528', 'chem_id': 'PROTON'}
{'reaction_id': 'RXN3DJ-35528', 'chem_id': 'CO-A'}
{'reaction_id': 'ACETYLSEROTONIN-O-METHYLTRANSFERASE-RXN', 'chem_id': 'PROTON'}
{'reaction_id': 'ACETYLSEROTONIN-O-METHYLTRANSFERASE-RXN', 'chem_id': 'ADENOSYL-
HOMO-CYS'}
{'reaction_id': 'ACETYLSEROTONIN-O-METHYLTRANSFERASE-RXN', 'chem_id':
'N-ACETYL-5-METHOXY-TRYPTAMINE'}
{'reaction_id': 'TRYPTOPHAN-5-MONOOXYGENASE-RXN', 'chem_id': '5-HYDROXY-
TRYPTOPHAN'}
{'reaction_id': 'TRYPTOPHAN-5-MONOOXYGENASE-RXN', 'chem_id': 'CPD-5881'}
```

1.1.4 Replace humancyc compound CPD-14053 with Ecocyc compound CPD0-2101 (MH4)

CPD0-1201 is a product of folM.

```
match(n:Compound {biocyc_id:'CPD-14053'}) detach delete n;
```

```
match(n:Compound {biocyc_id:'CPD0-2101'}), (r:Reaction {biocyc_id:'TRYPTOPHAN-5-MONOOXYGENASE-
```

Export database as json file:

```
call apoc.export.json.all("/Users/rcai/data/database_backup/ecocyc-lei.json", {useTypes: true})
```

1.2 Add phhB step

TpH: Tryptophan + MH4 -> 5-hydroxytryptophan + MH3OH
folM: qMH2 + NADPH + H = MH4 + NADP+ (EcoCyc)
phhB: MH3OH -> qMH2 + H2O (Reaction biocyc_id: RXN-7908)

- MH4: 5,6,7,8-tetrahydromonapterin (biocyc_id: CPD0-2101)
- MH3OH: 4a-hydroxytetrahydropterin (CPD-5881)
- qMH2: 7,8-dihydromonapterin (biocyc_id: CPD-11770)

set synonyms to easy read

```
match(n:db_BioCyc {biocyc_id: 'CPD0-2101'}) set n.displayName = 'MH4', n.synonyms = [n.name, n
match(n:db_BioCyc {biocyc_id: 'CPD-5881'}) set n.displayName = 'MH3OH', n.synonyms = [n.name, n
match(n:db_BioCyc {biocyc_id: 'CPD-11770'}) set n.displayName = 'qMH2', n.synonyms = [n.name, n
```

1.2.1 add reaction

```
merge (n:db_BioCyc:Reaction {biocyc_id: 'RXN-7908'})
set n.eid = n.biocyc_id, n.displayName='phhB RXN',
n.comment='knockin, modified', n.detail="MH3OH => qMH2 + H2O"
return n;

match (n:Reaction {biocyc_id: 'RXN-7908'}), (c:Compound {biocyc_id:'CPD-5881'})
merge (c)-[:CONSUMED_BY]->(n);

match (n:Reaction {biocyc_id: 'RXN-7908'}), (c:Compound {biocyc_id:'CPD-11770'})
merge (n)-[:PRODUCES]->(c) return n;
```

1.2.2 Change reaction displayName and detail

```
match(n:Reaction {biocyc_id:'TRYPTOPHAN-5-MONOOXYGENASE-RXN'})
set n.displayName='TpH RXN', n.comment="knockin, modified",
n.detail = "L-tryptophan + MH4 => 5-hydroxy-L-tryptophan + MH3OH" return n;

match(n:Reaction {biocyc_id:'RXN3DJ-170'})
set n.displayName = 'DDC RXN', n.comment='knockin' return n;

match(n:Reaction {biocyc_id:'RXN3DJ-35528'})
set n.displayName = 'AANAT RXN', n.comment = 'knockin' return n;

match(n:Reaction {biocyc_id:'ACETYLSEROTONIN-O-METHYLTRANSFERASE-RXN'})
set n.displayName='ASMT RXN', n.comment = 'knockin' return n;

match(n:Reaction {biocyc_id: 'RXN0-6367'})
set n.displayName = 'folM RXN',  n.detail = "qMH2 + NADPH + H+ => MH4 + NADP+"
return n;
```

1.2.3 set entityType

```
match(n:Compound) where not exists(n.entityType) set n.entityType = 'Chemical';

match(n:Reaction) where not exists(n.entityType) set n.entityType = 'Reaction';
```

2 Summary

2.0.1 Knock-out

```
TnaA:  ['EG11005', 'TRYPTOPHAN-MONOMER', 'TRYPTOPHAN-CPLX', 'TRYPTOPHAN-RXN']

yddG: ['EG12713', 'EG12713-MONOMER', 'TRANS-RXN0-265']

ldhA: ['G592', 'DLACTDEHYDROGNAD-MONOMER', 'CPLX0-8158', 'DLACTDEHYDROGNAD-RXN']

cytR: ['EG10200', 'PD04028', 'CPLX0-7740', 'RXN0-6409', 'CPLX0-8051']
```

2.0.2 Knock-in

- TpH: TRYPTOPHAN-5-MONOOXYGENASE-RXN
  - Tryptophan + MH4 -> 5-hydroxytryptophan + MH3OH
- DDC: RXN3DJ-170
  - 5-hydroxy-L-tryptophan + H+ => CO2 + serotonin

- AANAT: RXN3DJ-35528
  - acetyl-CoA + serotonin => N-acetyl-serotonin + H+ + coenzyme A
- ASMT: ACETYLSEROTONIN-O-METHYLTRANSFERASE-RXN
  - S-adenosyl-L-methionine + N-acetyl-serotonin => H+ + S-adenosyl-L-homocysteine + melatonin
- folM: RXN0-6367
  - qMH2 + NADPH + H = MH4 + NADP+ (EcoCyc)

- phhB: RXN-7908
  - MH3OH -> qMH2 + H2O

2.0.3 Naming

- MH4: 5,6,7,8-tetrahydromonapterin (biocyc_id: CPD0-2101)

- MH3OH: 4a-hydroxytetrahydropterin (CPD-5881)
- qMH2: 7,8-dihydromonapterin (biocyc_id: CPD-11770)

[ ]: