KEGGscape

Release 0.8.1

Oct 21, 2018

Contents

KEGGscape constructs KEGG pathway on Cytoscape3 (formerly known as KGMLReader for Cytoscape 2.*).
In contrast to KEGG web, you can edit the network and map your data as you like.

CHAPTER 1

Installing KEGGscape

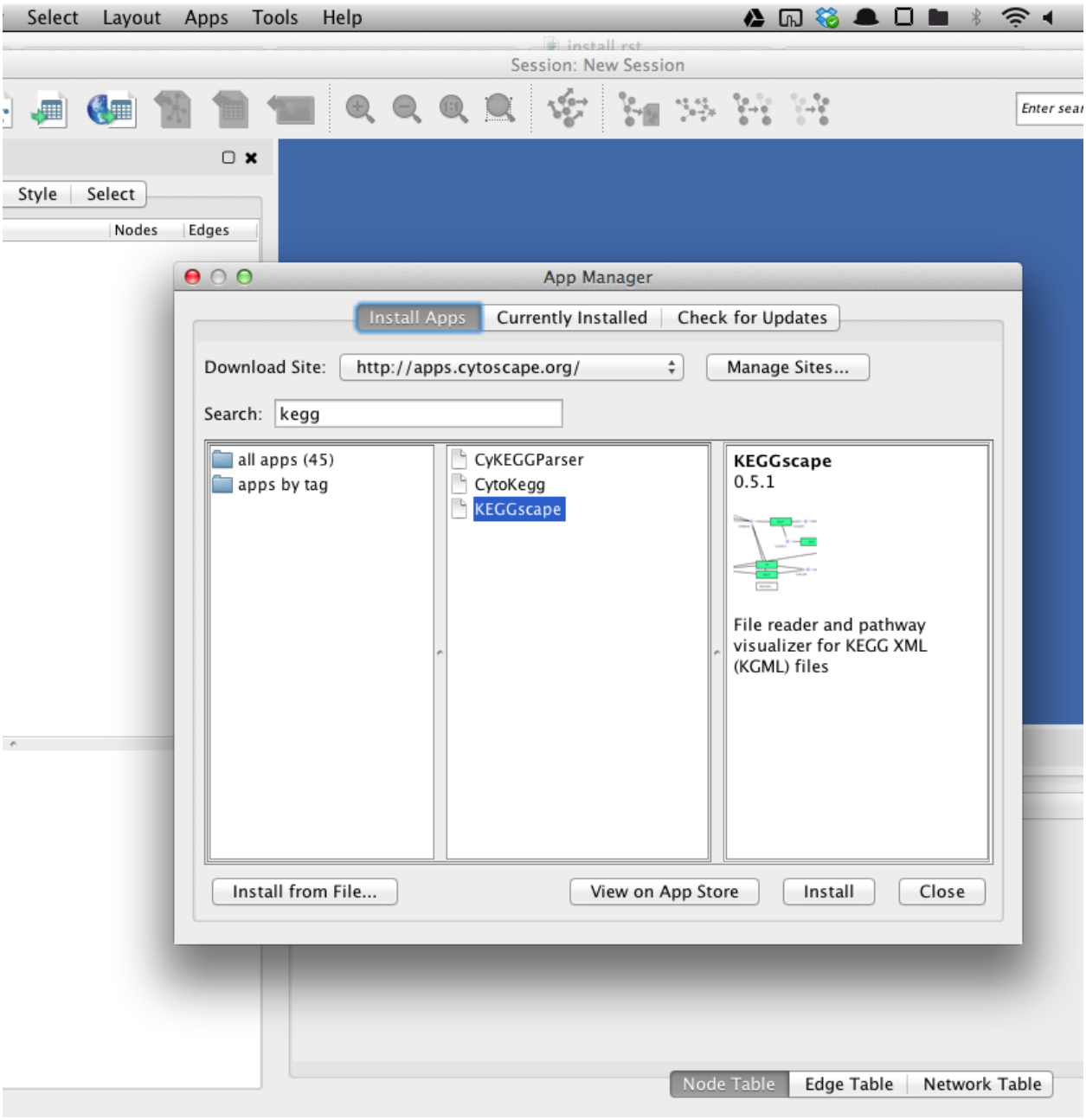
KEGGscape source code is distributed as open source software under the Apache License, Version 2.0 and is available at GitHub.

https://github.com/idekerlab/KEGGscape

KEGGscape requires Cytoscape version 3.6. First you need to download Cytoscape from

http://cytoscape.org/download.html

You can install KEGGscape with Cytoscape app manager.


Select
Layout
Apps
Tools
Help
Session: New Session
Style
Select
Nodes
Edges
App Manager
Install Apps
Currently Installed
Check for Updates
Download Site:
http://apps.cytoscape.org/
Manage Sites...
Search:
kegg
all apps (45)
apps by tag
CyKEGGParser
CytoKegg
KEGGscape
KEGGscape
0.5.1
File reader and pathway visualizer for KEGG XML (KGML) files
Install from File...
View on App Store
Install
Close
Node Table
Edge Table
Network Table

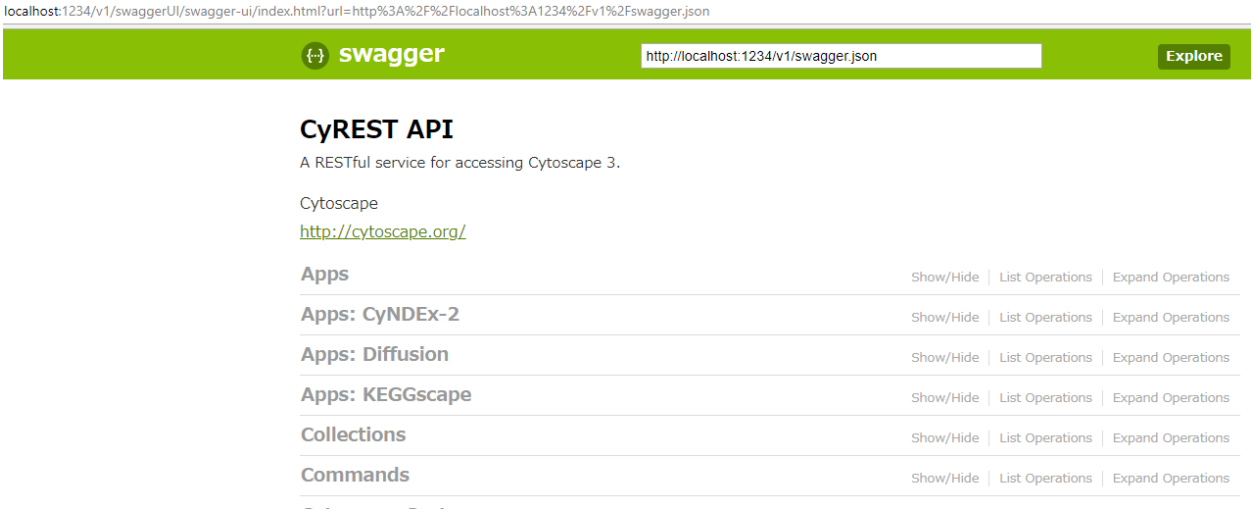
CHAPTER 2

How to import KEGG pathway xml(kgml) to Cytoscape

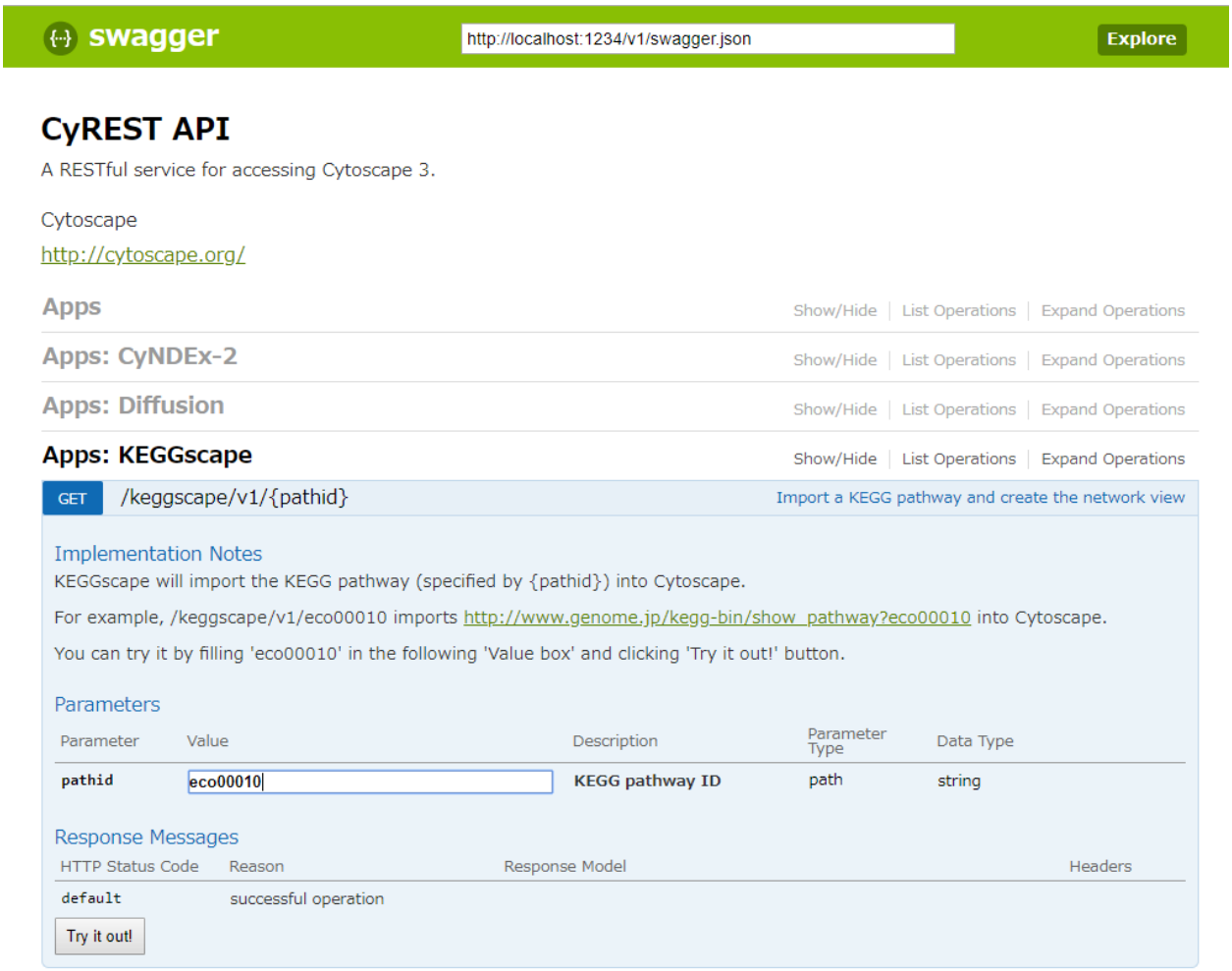
First we show how to import KEGG pathway xml(kgml) to Cytoscape.

2.1 Importing kgml to Cytoscape with REST endpoint

KEGGscape exposes a REST endpoint to directly import a KEGG pathway entry and it is documented in the main Swagger page generated by CyREST (available under: Help -> Automation -> CyREST API).



You can import kgml to Cytoscape with filling the KEGG pathway ID and clicking the “Try it out!” button.



2.2 Importing kgml to Cytoscape by manually downloading kgml

2.2.1 Downloading KEGG pathway kgml

You can download KEGG pathway kgml without opening web browser (if you know the KEGG pathway entryID you want to import).

```
wget http://rest.kegg.jp/get/eco00020/kgml -O eco00020.xml
```

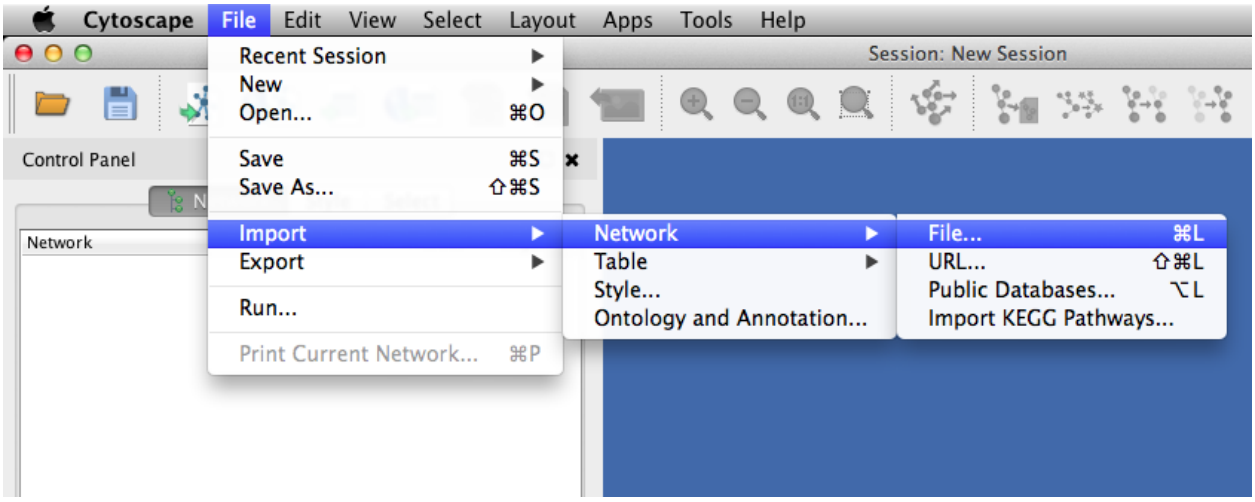
eco00020.xml is TCA cycle of Escherichia coli K-12 MG1655.

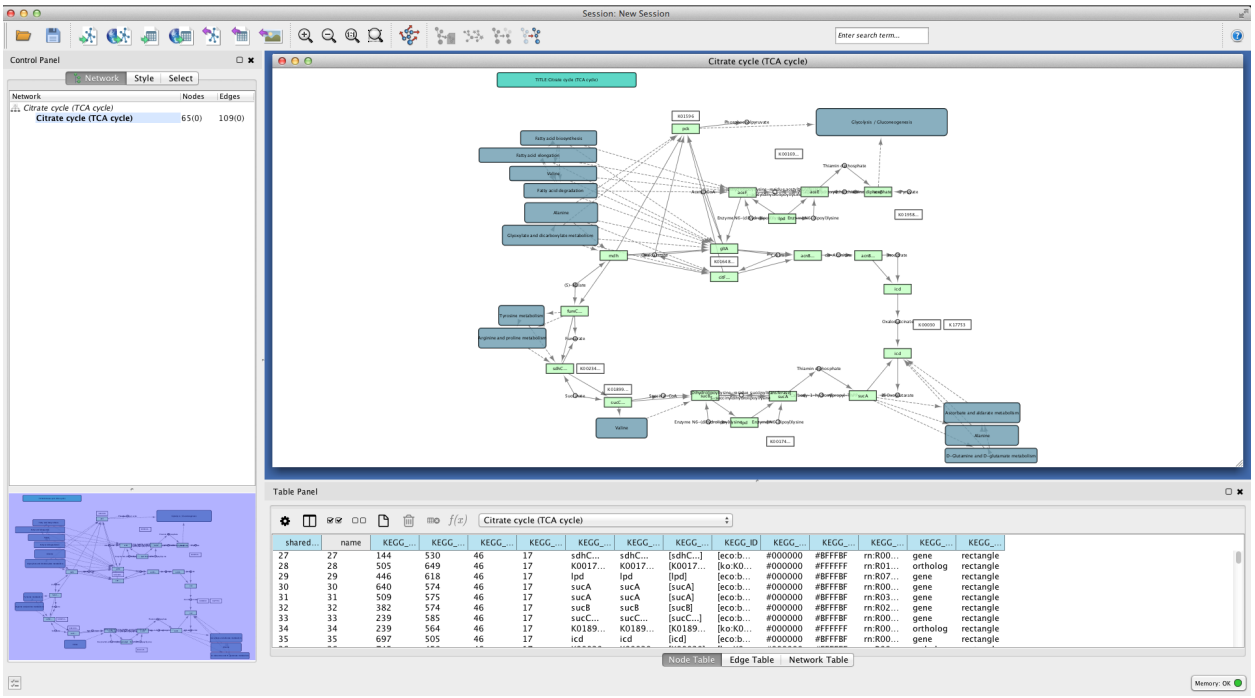
2.2.2 Importing kgml to Cytoscape by GUI

You can import kgml to Cytoscape from menu bar

File -> Import -> Network -> File

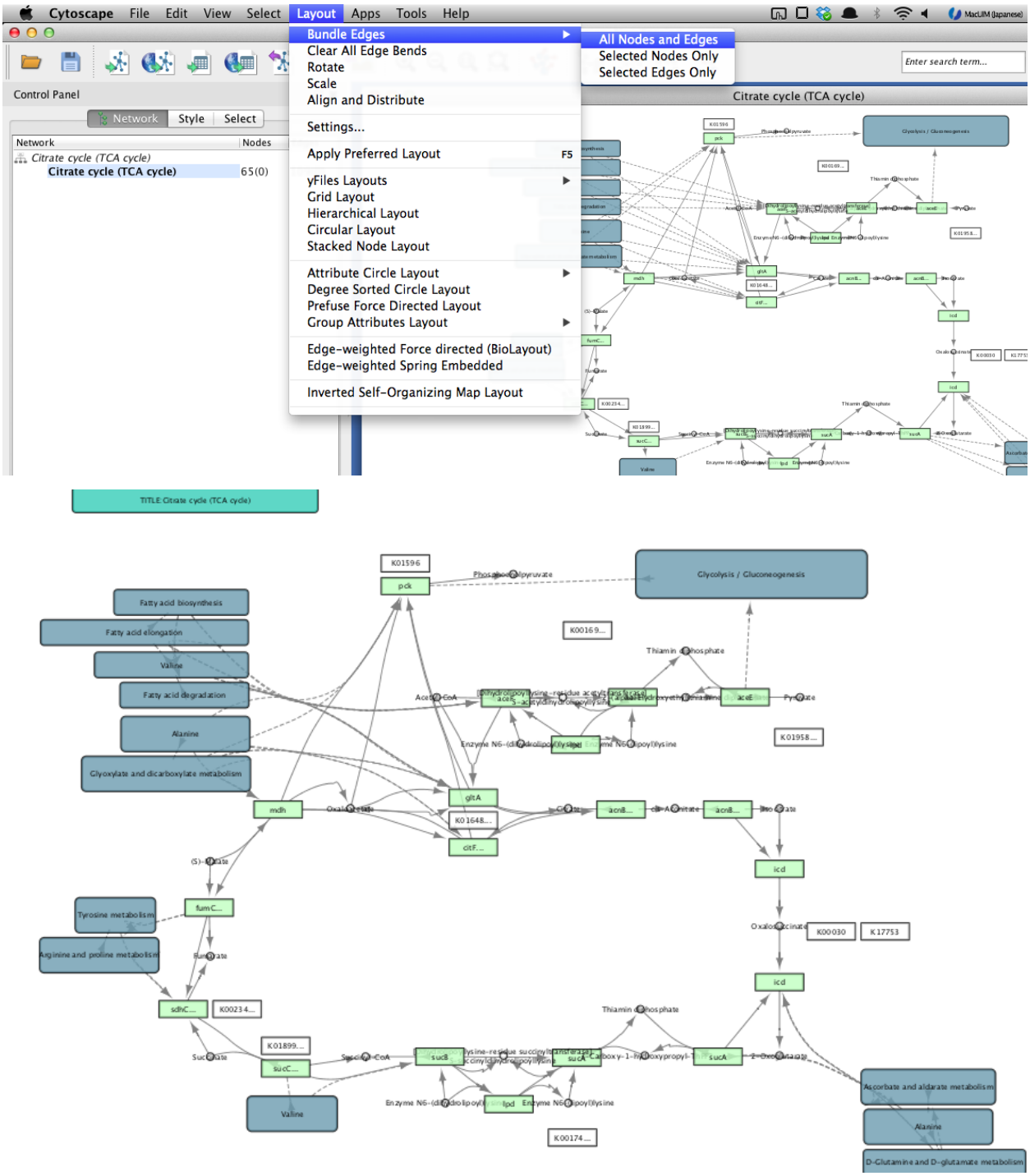
and open eco00020.xml.





2.3 How to bundle edges

KEGGscape creates two edges for a reversible reaction, if you want to bundle these reversible reactions like KEGG, please select “Bundle edges” from “Layout” menu.


Cytoscape File Edit View Select Layout Apps Tools Help
Bundle Edges
Clear All Edge Bends
Rotate
Scale
Align and Distribute
Settings...
Apply Preferred Layout F5
yFiles Layouts
Grid Layout
Hierarchical Layout
Circular Layout
Stacked Node Layout
Attribute Circle Layout
Degree Sorted Circle Layout
Prefuse Force Directed Layout
Group Attributes Layout
Edge-weighted Force directed (BioLayout)
Edge-weighted Spring Embedded
Inverted Self-Organizing Map Layout
All Nodes and Edges
Selected Nodes Only
Selected Edges Only
Control Panel
Network Style Select
Citrate cycle (TCA cycle)
Enter search term...
TITLE:Citrate cycle (TCA cycle)
Glycolysis / Gluconeogenesis
Fatty acid biosynthesis
Fatty acid elongation
Valine
Fatty acid degradation
Alanine
Glyoxylate and dicarboxylate metabolism
Tyrosine metabolism
Arginine and proline metabolism
Ascorbate and aldarate metabolism
D-Glutamine and D-glutamate metabolism
pck
mdh
gltA
acnB
icd
fumC
sdhC
sucC
sucB
lpd
sucA
aceE
K01596
K00169
K01958
K01648
K00030
K17753
K00234
K01899
K00174

CHAPTER 3

Combination of Python scripts and KEGGscape

Scripting language support is an experimental feature in Cytoscape 3.

Cytoscape 3 supports scripting language. Here we show a sample of Python + KEGGscape.

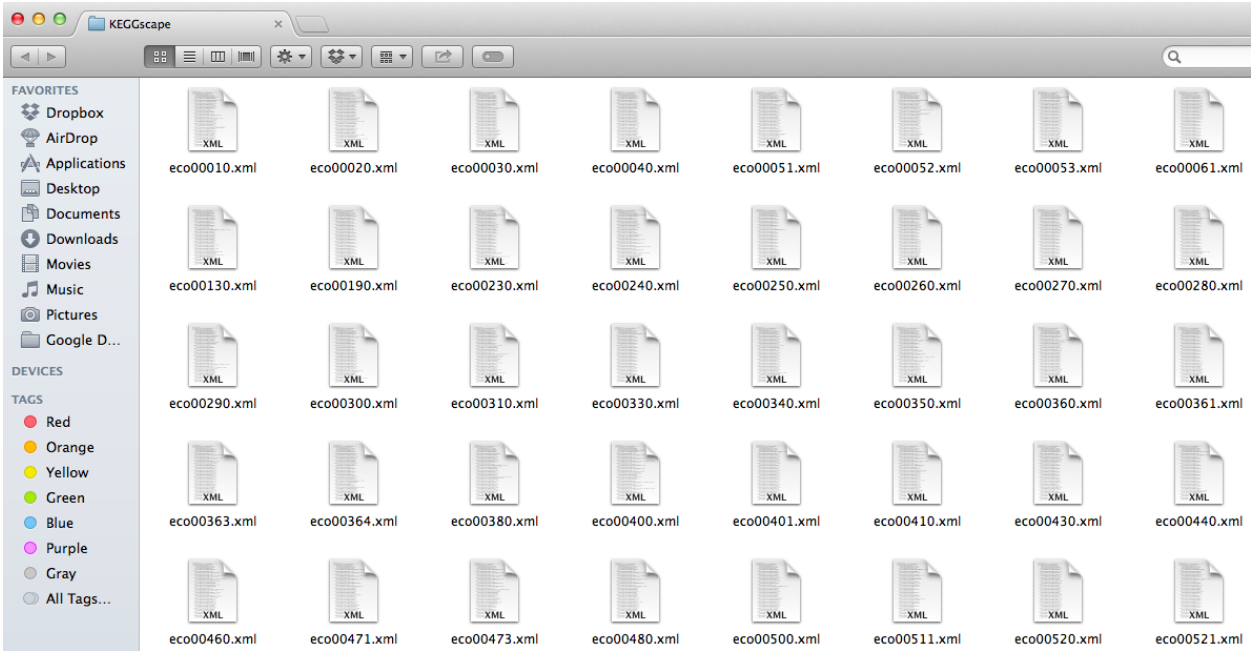
We import all Ecoli pathways to Cytoscape.

3.1 Importing all KEGG pathways of Escherichia coli K-12 MG1655

First we download all Ecoli pathways with the following Python script. This script requires requests Python package.

```
import requests

ORGANISM = "eco"

pathways = requests.get('http://rest.kegg.jp/list/pathway/' + ORGANISM)
for line in pathways.content.split('\n'):
    pathwayid = line.split('\t')[0].replace('path:', '')
    kgml = requests.get('http://rest.kegg.jp/get/' + pathwayid + '/kgml')
    f = open(pathwayid + '.xml', 'w')
    f.write(kgml.content)
    f.close
```

You will get all eco KGMLs like this.



Next we show a sample to batch-import kgml files from Python script. To use Python from Cytoscape3, you need to download jython-standalone from here , and move the jython-standalone like this.

Now you can batch-import kgml files with Python. Here we import all carbohydrate metabolism kgml files. (Of course you can import all pathways, but it takes time and cys file get so big.)

```
mkdir carbohydrate
mv eco00010.xml eco00020.xml eco00030.xml eco00040.xml eco00051.xml
eco00052.xml eco00053.xml eco00500.xml eco00520.xml eco00562.xml
eco00620.xml eco00630.xml eco00640.xml eco00650.xml eco00660.xml
```

next run cytoscape3, go in osgi shell and run following Python script(save as load_kegg.py).

```
from java.io import File

KEGG_DIR = "/ABS_PATH_TO/carbohydrate/"
pathways = ["eco00010.xml", "eco00020.xml", "eco00030.xml", "eco00040.xml", "eco00051.
↪xml", "eco00052.xml", "eco00053.xml", "eco00500.xml", "eco00520.xml", "eco00562.xml
↪", "eco00620.xml", "eco00630.xml", "eco00640.xml", "eco00650.xml", "eco00660.xml"]

loadNetworkTF = cyAppAdapter.get_LoadNetworkFileTaskFactory()
taskManager = cyAppAdapter.getTaskManager()

allTasks = None

for pathway in pathways:
    kgmlpath = File(KEGG_DIR + pathway)
    print str(kgmlpath)
    itr = loadNetworkTF.createTaskIterator(kgmlpath)
    if allTasks is None:
        allTasks = itr
    else:
        allTasks.append(itr)
```

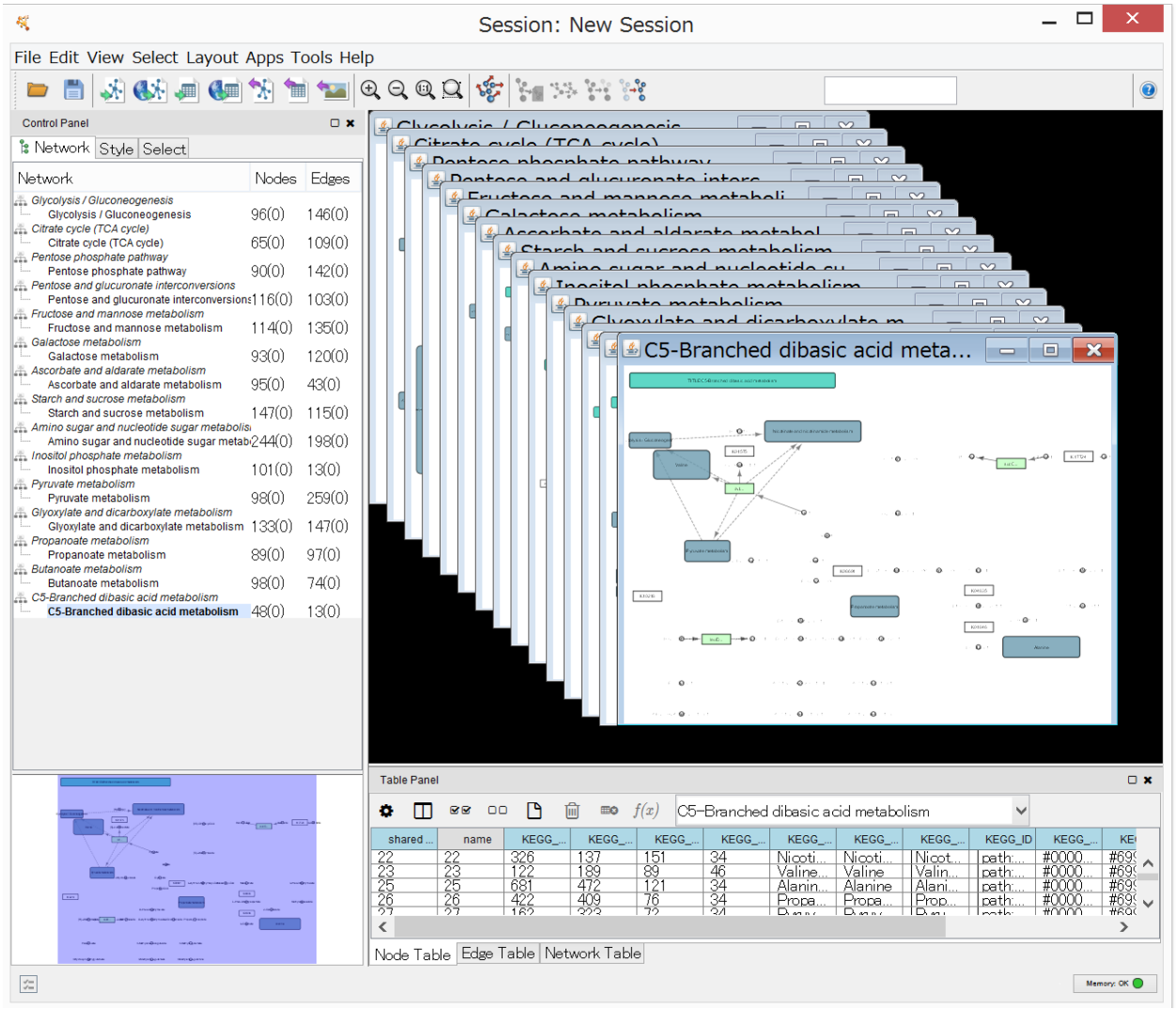
(continues on next page)

(continued from previous page)

```
taskManager.execute(allTasks)
```

Save this Python script as load_kegg.py. To run load_kegg.py, type (from Cytoscape3 OSGi shell)

```
cytoscape:script python /ABS_PATH_TO_SCRIPT/load_kegg.py
```



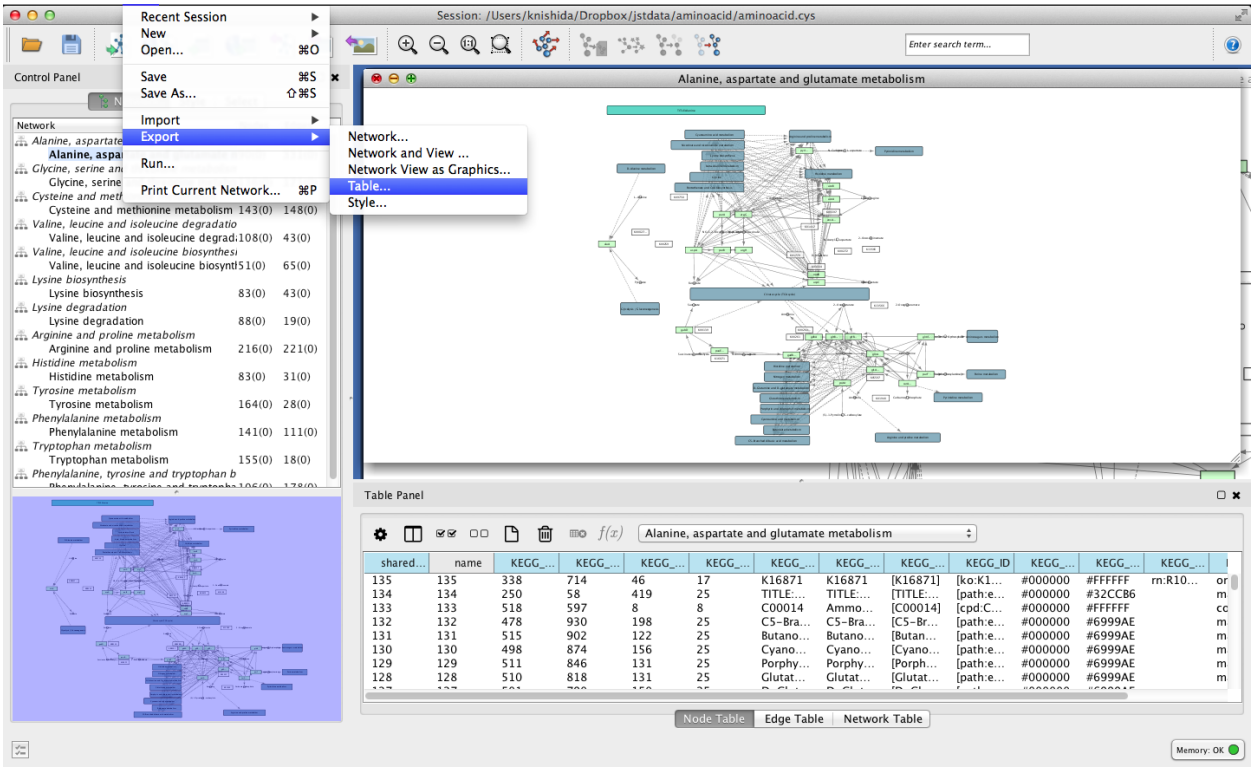
CHAPTER 4

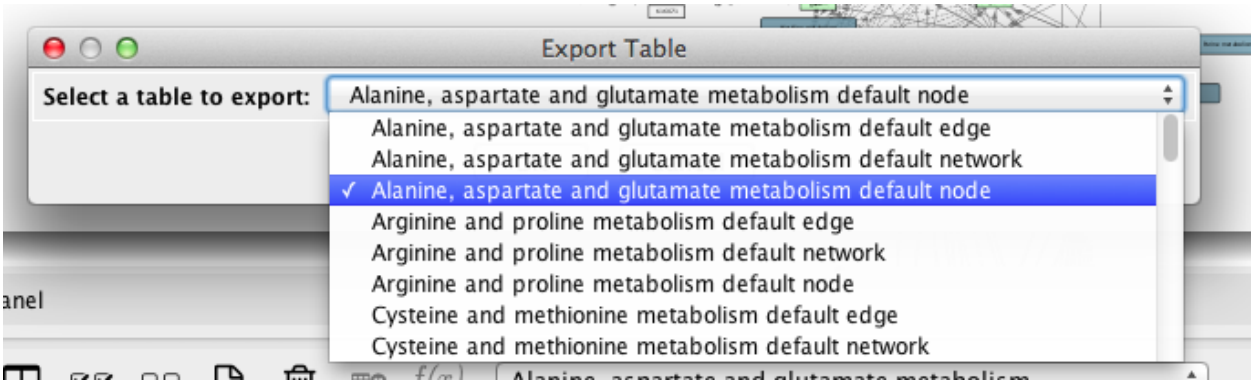
Mapping drug targets on KEGG pathway

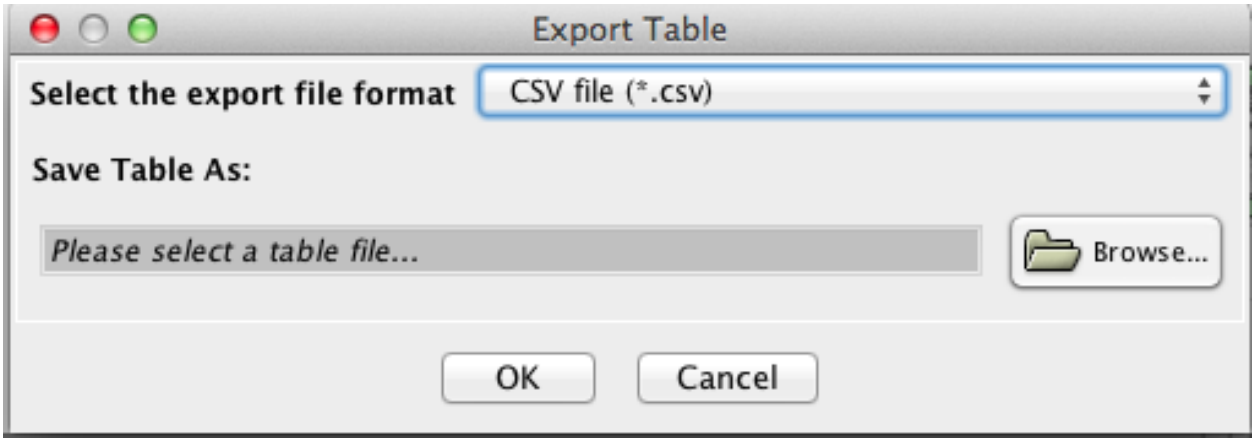
Here we show a example of data integration. We map drug targets(from Drugbank) on KEGG pathway. To manage several tables, we use MongoDB and PyMongo.

4.1 Importing all data into MongoDB

First we export node attribute table of *Alanine, aspartate and glutamate metabolism* as alanine_nodes.csv.







Next we download drug targets from Drugbank and id convert table with KEGG REST API.

```
wget http://www.drugbank.ca/system/downloads/current/all_target_ids_all.csv.zip
unzip all_target_ids_all.csv.zip
wget http://rest.kegg.jp/conv/eco/uniprot
mv uniprot conv_eco_uniprot.tsv
```

Finally we import these tables into mongodb.

```
mongoimport --db keggscape --collection alanine_nodes --headerline --type csv --file␣
↪alanine_nodes.csv
mongoimport --db keggscape --collection all_target_ids_all --headerline --type csv --
↪file all_target_ids_all.csv
mongoimport --db keggscape --collection conv_eco_uniprot -f uniprot_id,kegg_id --type␣
↪tsv --file conv_eco_uniprot.tsv
```

4.2 Merging tables with PyMongo

We integrate the three table(network nodes, drug targets table, id conversion table). Here we append columns *drug target* and *drug* to Cytoscape's node table.

```
from pymongo import MongoClient

client = MongoClient()
db = client['keggscape']
```

(continues on next page)

(continued from previous page)

```
node_collection = db['alanine_nodes']
drug_collection = db['all_target_ids_all']
conv_collection = db['conv_eco_uniprot']

gene_table = node_collection.find({"KEGG_NODE_TYPE": "gene"})

for genes in gene_table:
    locuses = genes["KEGG_ID"].split("\r") #newline character depends on your OS, I␣
↪exported cytoscape table on Mac
    for locus in locuses:
        ids = conv_collection.find_one({"kegg_id": locus})
        drug = drug_collection.find_one({"UniProt ID": ids["uniprot_id"].replace("up:
↪", "")})
        if drug != None:
            node_collection.update({"_id": genes["_id"]}, {"$push": {"drug_ids": drug[
↪"Drug IDs"], "target_id": drug["ID"], "target": locus}})
            node_collection.update({"_id": genes["_id"]}, {"$set": {"is_target": 1}})
```

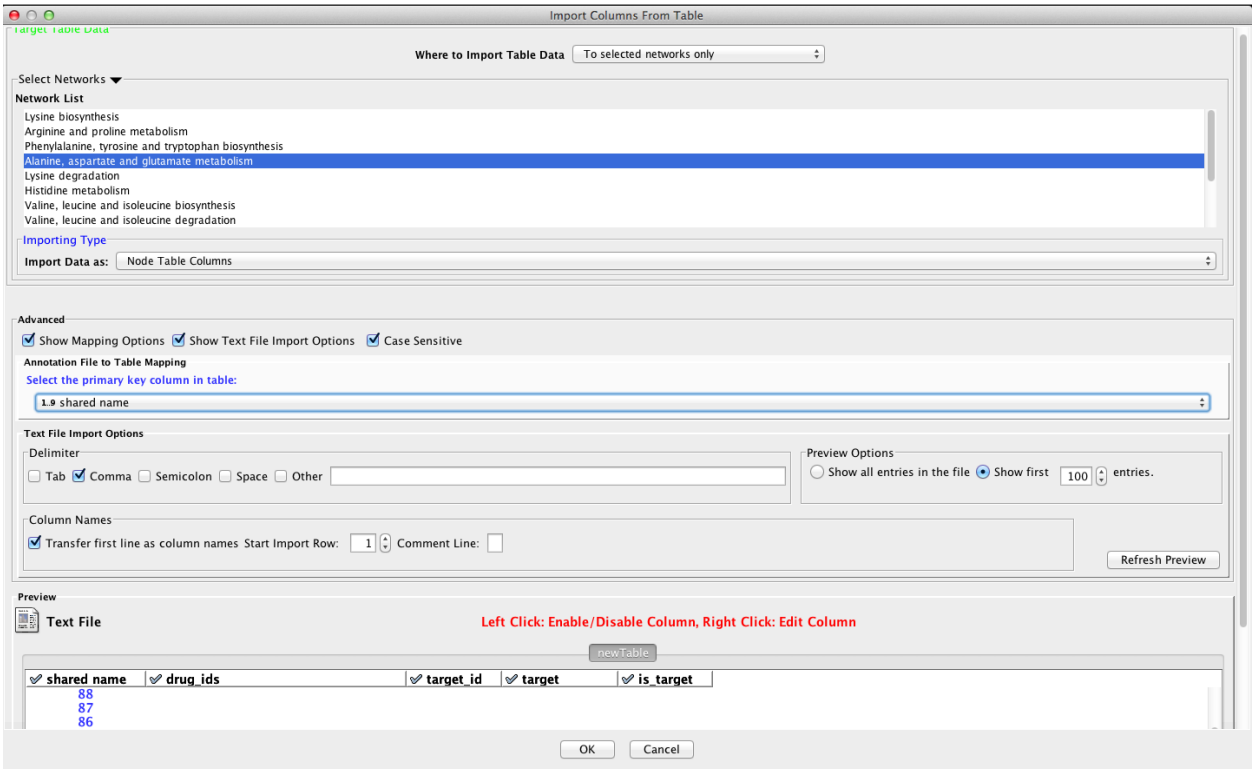
Next we create fields.txt to export the new node table.

```
shared name
drug_ids
target_id
target
is_target
```

and export node table as csv.

```
mongoexport --db keggscape --collection alanine_nodes --csv --fieldFile fields.txt --
↪out alanine_drugs.csv
```

import this alanine_drugs.csv into Cytoscape and highlight drug targets as below.


Import Columns From Table
Where to Import Table Data
To selected networks only
Select Networks
Network List
Lysine biosynthesis
Arginine and proline metabolism
Phenylalanine, tyrosine and tryptophan biosynthesis
Alanine, aspartate and glutamate metabolism
Lysine degradation
Histidine metabolism
Valine, leucine and isoleucine biosynthesis
Valine, leucine and isoleucine degradation
Importing Type
Import Data as:
Node Table Columns
Advanced
Show Mapping Options
Show Text File Import Options
Case Sensitive
Annotation File to Table Mapping
Select the primary key column in table:
1.9 shared name
Text File Import Options
Delimiter
Tab
Comma
Semicolon
Space
Other
Preview Options
Show all entries in the file
Show first
100
entries.
Column Names
Transfer first line as column names
Start Import Row:
Comment Line:
Refresh Preview
Preview
Text File
Left Click: Enable/Disable Column, Right Click: Edit Column
newTable
shared name
drug_ids
target_id
target
is_target
88
87
86
OK
Cancel

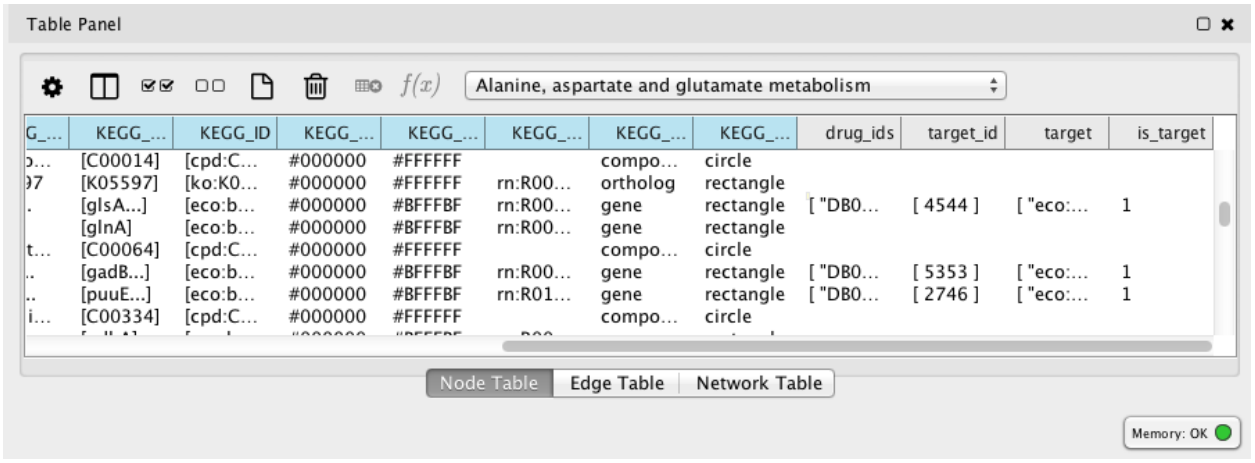
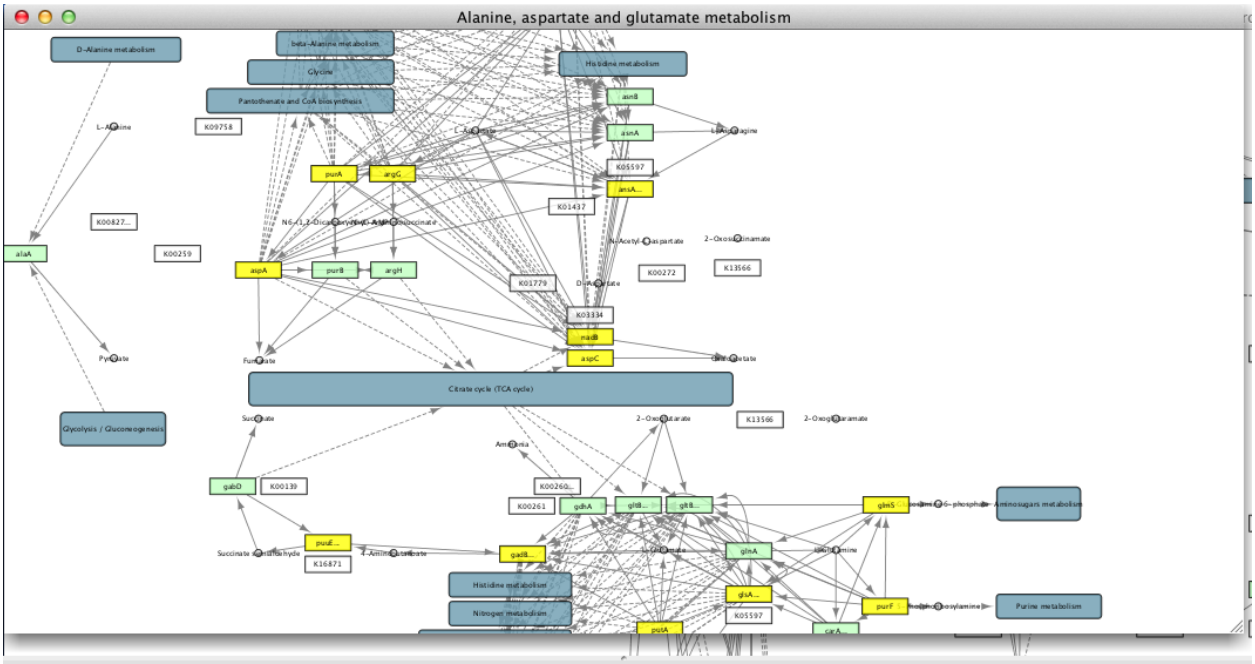


Table Panel
Alanine, aspartate and glutamate metabolism
KEGG_ID
drug_ids
target_id
target
is_target
Node Table
Edge Table
Network Table
Memory: OK



KEGG_...	KEGG_...	KEGG_...	KEGG_...	KEGG_ID	KEGG_...	KEGG_...	KEGG_...	KEGG_...	KEGG_...	drug_ids	target_id	target	△ is_target
17	argG	argG	[argG]	[eco:b...	#000000	#BFFFBF	rn:R01...	gene	rectangle	["DB0...	[3194]	["eco:...	1
17	putA	putA	[putA]	[eco:b...	#000000	#BFFFBF	rn:R00...	gene	rectangle	["DB0...	[3254]	["eco:...	1
17	gadB...	gadB...	[gadB...]	[eco:b...	#000000	#BFFFBF	rn:R00...	gene	rectangle	["DB0...	[5353]	["eco:...	1
17	puuE	puuE	[puuE...]	[eco:b	#000000	#BFFFBF	rn:R01	gene	rectangle	["DB0	[2746]	["eco:	1

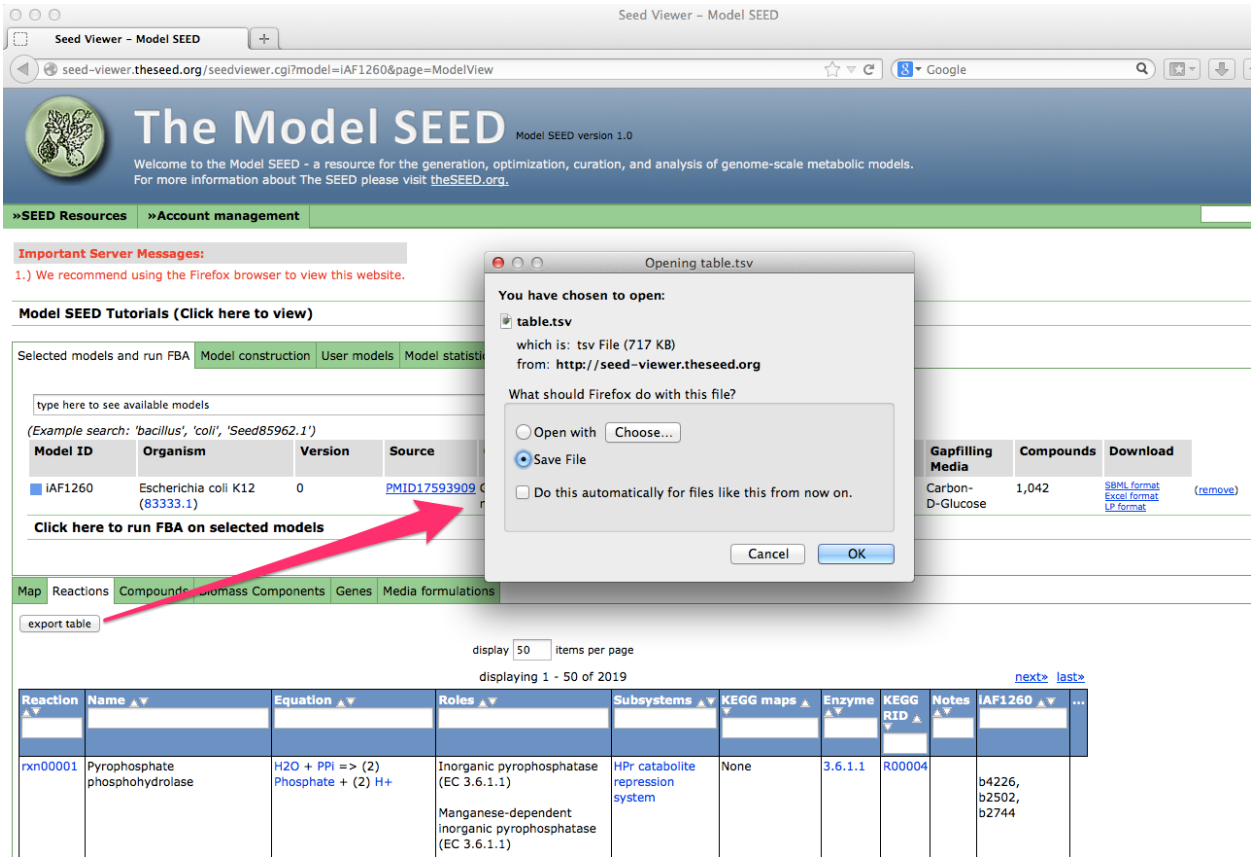
CHAPTER 5

Mapping genome scale metabolic model on KEGG pathway

Here we show the other example of data integration. We map iAF1260(a genome-scale metabolic reconstruction for Escherichia coli K-12 MG1655 that accounts for 1260 ORFs) on KEGG pathway.

5.1 Importing iAF1260 into MongoDB

You can download iAF1260 reaction table from ModelSEED.



and import this table into MongoDB, like this.

```
mongoimport --db keggscape --collection iaf1260 --type tsv --headerline --file table.
↪tsv
```

and export Galactose metabolism pathway from Cytoscape and import it like this.

```
mongoimport --db keggscape --collection galactose_node --headerline --type csv --file␣
↪galactose_node.csv
```

This Python script append column which enzyme genes differ between KEGG and iAF1260.

```
from sets import Set
from pymongo import MongoClient

client = MongoClient()
db = client['keggscape']

node_collection = db['galactose_node']
model_collection = db['iaf1260']

kegggene_table = node_collection.find({"KEGG_NODE_TYPE": "gene"})
modelreaction_table = model_collection.find({"KEGG RID": {"$regex": "R[0-9]{5}"}})

for kegggene in kegggene_table:
    kegggenes = kegggene['KEGG_ID'].split("\r")
    keggonly_genes = []
    modelonly_genes = []
```

(continues on next page)

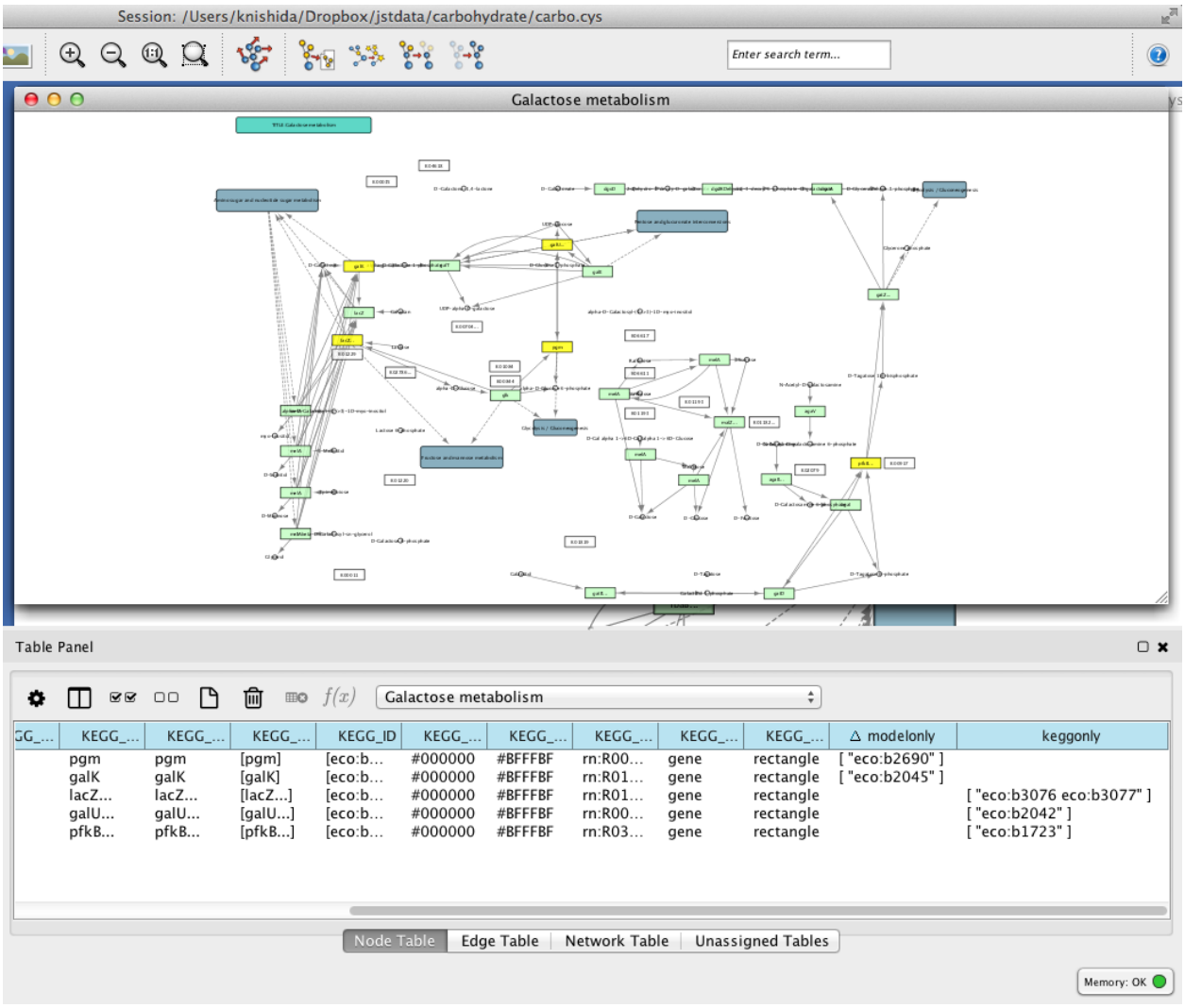
(continued from previous page)

```
    for keggreaction in kegggene['KEGG_NODE_REACTIONID'].split(" "):
        modelkeggreaction_table = model_collection.find({"KEGG RID": {"$regex":␣
↪keggreaction.replace("rn:", "")}})

        if modelkeggreaction_table.count() > 0:
            for modelkeggreaction in modelkeggreaction_table:
                modelgenes = modelkeggreaction['iAF1260\r'].strip().replace("<br>",
↪"eco:").split(", ")

                if Set(kegggenes) != Set(modelgenes):
                    keggonly = Set(kegggenes) - Set(modelgenes)
                    modelonly = Set(modelgenes) - Set(kegggenes)
                    if len(keggonly) > 0:
                        node_collection.update({"_id": kegggene["_id"]}, {"$push": {
↪"keggonly": " ".join(keggonly)}})
                    else:
                        node_collection.update({"_id": kegggene["_id"]}, {"$push": {
↪"modelonly": " ".join(modelonly)}})
```

And export galactose_node collection and reimport to Cytoscape.

```
mongoexport --db keggscape --collection galactose_nodes --csv --fieldFile genediff_
↪fields.txt --out new_galactose_nodes.csv
```

Here is the annotation difference between iAF1260 and KEGG



This work was supported by the National Bioscience Database Center(NBDC) program *Database Integration Coordination Program (Tool Prototype for Integrated Database Analysis)*

CHAPTER 6

index

- genindex
- modindex
- search