Kgml2Jason

Kgml2Jason will grab genes' details in different pathways, which from KEGG with KEGG Makeup Language (KGML) file and export genes list and relationship of genes. The goal here is to visualize the pathway rebuild it in the level of genes.

Plugin Scripts
4.1 convert.py

4.1.1 Internal operation
First , this utility will use the parameter –f to determine the specified file and read it in. The output file will use the file name that put forward.
Second, this utility convert KGML file into JSON and grab the information of genes, such as the reactions of genes and the relationship between genes.
Third, this utility continue to integrate the information above into two files, gene.json and relation.json, which can be use directly in rewrite gene's pathway.
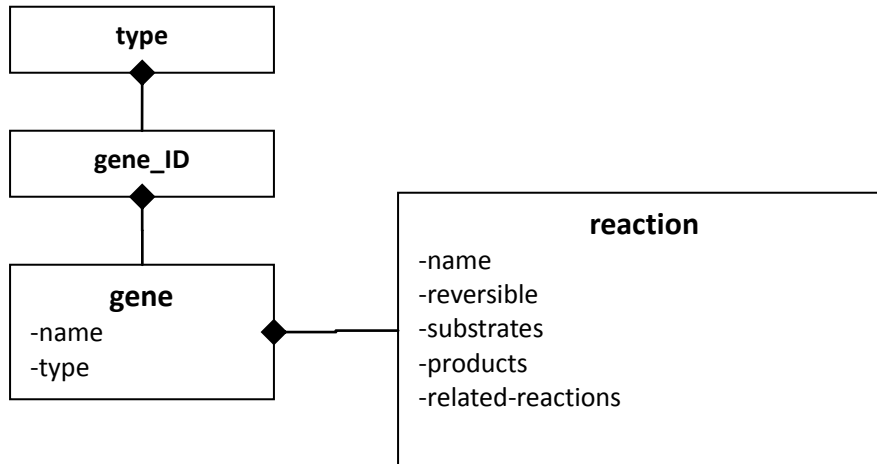
4.1.2 Example
python convert.py –f sce04111

4.1.3 Parameters:

		default
-f/--file	read kgml from FILENAME(omit '.xml'), produce two files: genelist and relation	

The format of output:
The output file is stored in /the path where you running this program.

_gene.json

```
            ┌─────────────────┐
            │      type        │
            └────────◆────────┘
                     │
            ┌─────────────────┐
            │     gene_ID      │
            └────────◆────────┘
                     │
            ┌─────────────────┐          ┌──────────────────────────┐
            │      gene        │          │        reaction          │
            │  -name           │◆─────────│  -name                   │
            │  -type           │          │  -reversible             │
            └─────────────────┘          │  -substrates             │
                                          │  -products               │
                                          │  -related-reactions      │
                                          └──────────────────────────┘
```

type

| ortholog	KO (ortholog group)
enzyme	Enzyme
reaction	Reaction
gene	gene product (mostly a protein)
group	a complex of gene products (mostly a protein complex)

geneID:the unique identification of gene

name	the KEGGID of this gene
type	the type of this gene

reaction

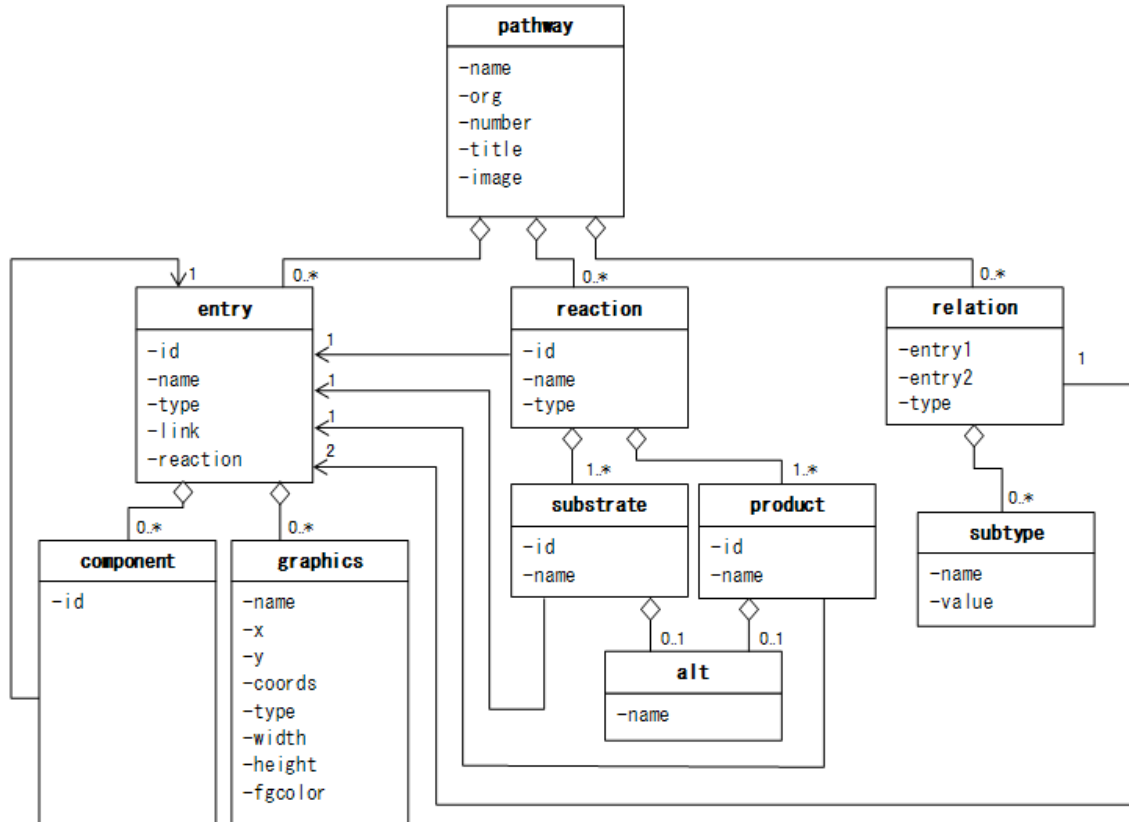| name	the KEGGID of this reaction
reversible	true: reversible reaction; false: irreversible reaction
substrates	KEGGID of substrate node
products	the KEGGID of product node
related-reactions	relate to another pathway or gene

_relation.json
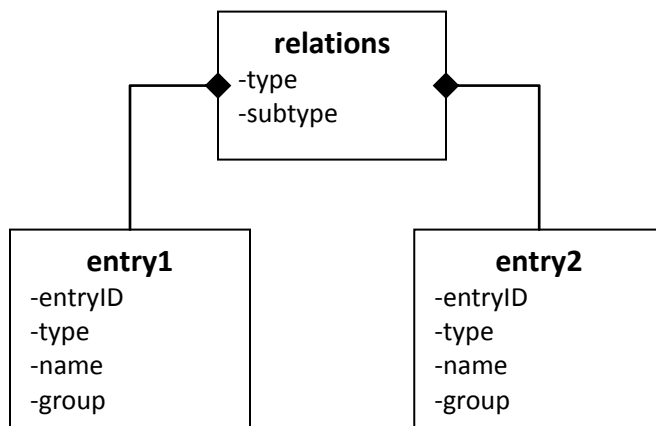
relations

type	the type of this relation[ECrel,PPrel.GErel,PCrel,maplink]
subtype	Interaction/relation information[activation/inhibition]
entry1	the first (from) entry that defines this relation
entry2	the second (to) entry that defines this relation

entry1&2

entry ID	the geneID of node which takes part in this relation
type	Have only two options: [gene/group]
name	the KEGGID of this gene
group	the node is a complex of gene products (mostly a protein complex)

4.1.4 How to finish this plunin?

KEGG is a database resource for understanding high-level functions and utilities of the biological system, and KGML is an XML presentation of the KEGG pathway database, which enables automatic drawing of KEGG pathways and provides facilities for computational analysis and modeling of gene/protein networks and chemical networks. Here is the data structure of KGML.

pathway

-name
-org
-number
-title
-image

entry

-id
-name
-type
-link
-reaction

reaction

-id
-name
-type

relation

-entry1
-entry2
-type

component

-id

graphics

-name
-x
-y
-coords
-type
-width
-height
-fgcolor

substrate

-id
-name

product

-id
-name

subtype

-name
-value

alt

-name

It's really complex and it will bother you to understand the kgml file! Do not worry, I will show you how to understand a KGML file and then how to convert it into JSON.

First, how to find or download KGML file?
Method:
download KGML" link for each pathway map.
If you choose a pathway with prefix "map", you cann't find the download link in the page,
that's because it can be generating almost ko/rn/ec/org files.
Such as "map04111", it has no link for download. But if you change the "map" into "sce" in
the URL, you can get the file.
KEGG API: http://www.kegg.jp/kegg/rest/keggapi.html
Take "sce04111" as an example, you can download the KGML file via the following web link:
http://rest.kegg.jp/get/sce04111/kgml

Second, the KGML file is difficult to find out the relationship through the data structure show above.
Here is the simple but straightforward tutorial to teach you how to understand a KGML file.
Take "sce04111.xml" as an example. We can simplify the data structure as:

-pathway

-entry (-component, -graphics)

-reaction (-substrate, -product, -alt)

-relation (-subtype)

The entry element can be path/ko/ec/rn/cpd/gl/org/group, enzyme/protein/gene will have relation and gene will also have compound and reactions. We choose one relation to be example:

```
<relation entry1="81" entry2="118" type="PPrel">
     <subtype name="activation" value="--&gt;"/>
</relation>
<entry id="81" name="sce:YCL061C" type="gene"
     link="http://www.kegg.jp/dbget-bin/www_bget?sce:YCL061C">
     <graphics name="MRC1, YCL060C" fgcolor="#000000" bgcolor="#BFFFBF"
          type="rectangle" x="1268" y="199" width="46" height="17"/>
<entry id="118" name="sce:YPL153C" type="gene"
     link="http://www.kegg.jp/dbget-bin/www_bget?sce:YPL153C">
     <graphics name="RAD53, LSD1, MEC2, SPK1" fgcolor="#000000" bgcolor="#BFFFBF"
          type="rectangle" x="1236" y="235" width="46" height="17"/>
</entry>
```

It means that gene(YCL061C) have activation effect in gene(YPL153C).By the way, through the graphics elements, we can definitely rewrite the connection between these two genes.

Third, thought we know how to get KGML file and can understand it but the crucial problem is, how to grab the information of genes and convert it into JSON.
Here we use Python programming Language, install the library "lxml" for processing XML and HTML, and we import json library for convert.

Fourth, you may asking: If there have any other software to do such job, that is read KGML files, convert it and rewrite it?
Of course, and we indeed tried but failed for it's not open source or the original source is difficult for modification.
Actually, if you want to do some visualize pathway job, Cytoscape is a good choice and it also have cytoscape.js for drawing.

Finally, we indeed done plentiful preparatory work, maybe in the end it's not useful in this software but it can expand our horizons.