COMP150CSB: Computational Systems Biology
Tufts University, Department of Computer Science
Prof. Soha Hassoun
rev 1

Homework 1: Accessing Data from the KEGG Database to Analyze RClass

---

Assignment Overview
We learned from class and our readings that KEGG is a large database that catalogues data about metabolic pathways, including reactions, compounds, and modules in various hierarchies. The RClass is relatively new and improves over the RPAIR classification by grouping similar molecular transformations in the same class.

In this homework we will analyze three different RClasses, each having distinct characteristics:

- RClass 00184, whose transformation pattern either adds or removes an S-CoA group, for the most part
- RC00300, which transforms compound C00033 (Acetate) to many other compounds
- RC00099, which transforms atom type C1a to C4a, without adding or removing functional groups or atoms during the transformation. RC00099 is a transformation pattern common among various substrate-product pairs

You will write various pieces of code that will allow generating a summary regarding each of the three RClasses. The summary should resemble the listing below. Additionally, you will analyze the data that you generate and provide a summary.

Reactions	Catalyzing Enzymes	RPAIRS	K Numbers	Names of K
R01277	1.2.1.42	C00154_C00517		
R02620	1.2.1.50	C00609_C02843	K03400	long-chain-fatty-acyl-CoA reductase
R10549	1.2.1.50	C00609_C03371	K13922, K18366	propionaldehyde dehydrogenase, acetaldehyde/propanal dehydrogenase
R01173	1.2.1.57	C00136_C01412	K00132, K04072, K04073, K18366	...
R01172	1.2.1.57, 1.2.1.10	C00136_C01412	...	..
R00740	...	...	..	...
...	...	...	...	...

Background
KEGG API. The KEGG database provides a RESTful interface. Please see
http://www.kegg.jp/kegg/rest/keggapi.html

KEGG Package in Biopython.org. The Biopython package offers several functions to work with the KEGG database. The code provides an interface to the REST-style API with the info, list, find, get, conv (convert from/to outside identifications), and provides functions to work with compounds and enzymes databases within KEGG. Please see http://biopython.org/DIST/docs/api/Bio.KEGG-module.html
As mentioned in class, there were bugs in this code due to the reformatting of the KEGG database. It seems that they fixed all issues in the Dec 2018 release of the biopython package.

Working with KEGG. It seems that bioptyhon limits KEGG access to 3 requests per second. If you find this slow, you may cache retrievals in local files to prevent repeated calls to the REST API.

Assignment Details
To help you write modular code that can be used later in the semester, and to help you debug and write better code, please implement the following functions.

1. Write a function, get_KEGG_data, that uses either the KEGG REST API or Requests library in python to retrieve a record and store it locally. Test that this function will execute correctly for a "get" operation for a compound, enzyme, or RClass. The function will either retrieve a new entry from KEGG and write it out, or perform no operation if the entry is already retrieved and stored locally. This function should return the name of the file that has the appropriate data. For example, get_KEGG_data('cpd', 'C00154') should return a string, "C00154.txt", the name of the file that stores the KEGG entry for compound C00154. The return string could also include the directory name, if appropriate. Cached files should be stored locally and should have the correct data.

2. Create a class called Reaction, similar to the KEGG Compound and Enzyme classes. Write a function to parse a KEGG reaction entry. This function should be modeled after the KEGG compound and enzyme classes, but customized to the fields associated with the Reaction entries. You do not have to implement **all** the fields (e.g., pathway, molecular weight, etc.) – just the ones necessary to finish this assignment. Test that your function works correctly for some of the reactions listed in the table above.

3. Create a class called RClass, similar to the KEGG compound and enzyme classes. Write a function to parse a KEGG RClass entry. This function should be modeled after the KEGG compound and enzyme classes, but customized to the fields associated with the RClass. You do not have to implement **all** the fields (e.g., pathway, molecular weight, etc.) – just the ones necessary to finish this assignment. Test that your function works correctly for the three RClasses.

4. Write a function, analyze_RClass, that is a method for the RClass class, that takes the name of an RClass, and generates the cross-referenced data as shown in the table above. This method should utilize the get_KEGG_data function, and the parse commands for the various reaction, enzyme, and compound classes as needed. The return should be a data structure with all the listed fields (reaction, enzymes, reactant-pair, Orthology (K numbers and K names)). Provide a function that can print the returned data formatted in a reasonable way.

5. Write a paragraph that discusses how the results for the three RClass are different or similar in terms of similar acting enzymes, or similar substrate and products that are associated with each class, and if the reactions are catalyzed by the same orthologs.

Submission
All homework submissions should be through Gradescope

All your code should be well documented, and you should have a "README" file that explains how one could run your code to generate the results. You should also provide a specific listing of your environment. If you are using anaconda, you can do that as follows:

```
conda list --explicit > name_environment.txt
```

All discussion/questions should be provided in a .txt or .pdf form.

All files needed to run your code should be submitted.