COMP 150 CSB –
Computational Systems Biology

Accessing & Analyzing Data In the KEGG Database

Soha Hassoun

Department of Computer Science (primary)
Department of Chemical and biological Engineering
Department of Electrical and Computer Engineering

Tufts UNIVERSITY

Outline

- Accessing KEGG Data
  - FTPd files
  - REST API
- Biopython KEGG package
  - Use requests instead of KEGG API (demo)
  - Let's look at the KEGG.Compound class
    - Works, but there are some bugs in the package, dated 2000
- Homework 1

Tufts pays for KEGG subscription

- We FTP and download the various databases
- Organization
  - Different databases (e.g. Compound, RCLASS, Enzyme, etc.) in tarred file
  - Large (eg. 140MB of text for enzymes)
- Useful for larger scale analysis

KEGG provides an automated way of retrieving the KEGG entries

- Provide a URL string requesting the data
  - Listing
    - List all organisms: http://rest.kegg.jp/list/organism
  - Finding entries with matching queries
    - http://rest.kegg.jp/find/rclass/00001
  - Get to retrieve given database entry
    - http://rest.kegg.jp/get/cpd:C01290
  - Link named fields in database entry
    - Link all EC numbers in map10: http://rest.kegg.jp/link/ec/map00010

- It is called a REST API
  - ReST: Representational State Transfer
  - Protocol to "get" data from a server
  - http://www.kegg.jp/kegg/docs/keggapi.html

About KEGG API

- URL Form
- Database name
- KEGG database entry Format
  - http://www.kegg.jp/kegg/docs/dbentry.html
- Let's look closely at FIND

- Most common complaint:
  - "I can't get this data"
    - Yes, indeed, you cannot

FIND

FIND

Name

find – find entries with matching query keyword or other query data

URL form

```
http://rest.kegg.jp/find/<database>/<query>

<database> = pathway | brite | module | ko | genome | genes | <org> | vg | ag |
             ligand | compound | glycan | reaction | rclass | enzyme | network |
             variant | disease | drug | dgroup | environ | <medicus>
```

```
http://rest.kegg.jp/find/<database>/<query>/<option>

<database> = compound | drug
<option> = formula | exact_mass | mol_weight
```

Description

This is a search operation. The first form searches entry identifier and associated fields shown below for matching keywords.

Database	Text search fields (see flat file format)
pathway	ENTRY and NAME
module	ENTRY and NAME
ko	ENTRY, NAME and DEFINITION
genome	ENTRY, NAME and DEFINITION
genes (<org>, vg, ag)	ENTRY, ORTHOLOGY, NAME and DEFINITION
compound	ENTRY and NAME
glycan	ENTRY, NAME and COMPOSITION
reaction	ENTRY, NAME and DEFINITION
rclass	ENTRY and DEFINITION
enzyme	ENTRY and NAME
network	ENTRY and NAME
variant	ENTRY and NAME
disease	ENTRY and NAME
drug	ENTRY and NAME
dgroup	ENTRY and NAME
environ	ENTRY and NAME

Keyword search against brite is not supported. Use /list/brite to retrieve a short list.

In the second form the chemical formula search is a partial match irrespective of the order of atoms given. The exact mass (or molecular weight) is checked by rounding off to the same decimal place as the query data. A range of values may also be specified with the minus(-) sign.

Examples

/find/genes/shiga+toxin	for keywords "shiga" and "toxin"
/find/genes/"shiga toxin"	for keywords "shiga toxin"
/find/compound/C7H10O5/formula	for chemical formula "C7H10O5"
/find/compound/O5C7/formula	for chemical formula containing "O5" and "C7"
/find/compound/174.05/exact_mass	for 174.045 =< exact mass < 174.055
/find/compound/300-310/mol_weight	for 300 =< molecular weight =< 310

Biopython KEGG package

- "open source international collaboration of volunteer developers, providing Python libraries for a wide range of bioinformatics problems"
- http://biopython.org/
- Limited implementation of KEGG parsing functions
  - Ahm.. and buggy
    - Beware of compound and enzyme parsing old format for KEGG for keyword pathway among other issues

Cock, P. J., Antao, T., Chang, J. T., Chapman, B. A., Cox, C. J., Dalke, A., ... & De Hoon, M. J. (2009). Biopython: freely available Python tools for computational molecular biology and bioinformatics. Bioinformatics, 25(11), 1422-1423.

Let's take a look at the compound parsing file

- Where is it?
  - Where you installed the bio package ☺
    - Look for Compound directory, and __init__.py
- Does the class definition conform to current compound entries in KEGG?
  - Compare
    - http://rest.kegg.jp/get/cpd:C01290
    - Record() class
- What is the return value for parse?
- Try two different access methods (request, and API)

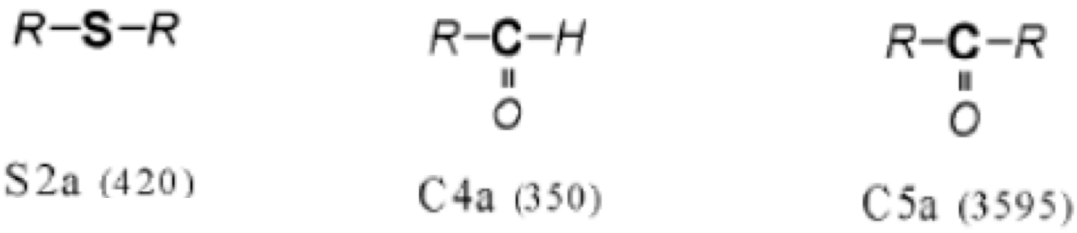
Let's look at kegg_get()

- Makes a call to _q with the correct arguments

Homework 1

- Let's analyze characteristics of three RClasses
- We would like to learn how the RClasses are similar or different
- Use some existing and new KEGG classes/functions to parse the data

RC00184



KEGG

RCLASS: RC00184

Help

Entry	RC00184 RClass
Definition	C4a-C5a:*-S2a:C1b+O4a-C1b+O5a
	
Reactant pair	C00083_C00222 C00091_C00232 C00100_C00479 C00136_C01412 C00154_C00517 C00593_C19685 C00609_C02843 C00609_C03371 C00609_C20683 C04076_C05535 Path search

KEGG **RCLASS: RC00184**

Help

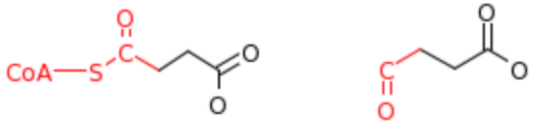
Entry	RC00184 RClass
Definition	C4a-C5a:*-S2a:C1b+O4a-C1b+O5a
	
Reactant pair	C00083_C00222 C00091_C00232 C00100_C00479 C00136_C01412 C00154_C00517 C00593_C19685 C00609_C02843 C00609_C03371 C00609_C20683 C04076_C05535

RC00184: cleaves S and beyond subgroups: Red boxed ones are very similar (with S-CoA).
Other two are not.

C00083 Malonyl-CoA
C00222 3-Oxopropanoate

C00083 C00222

C00091 Succinyl-CoA
C00232 Succinate semialdehyde



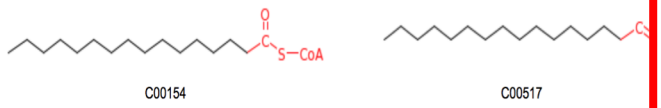
C00091 C00232

C00100 Propanoyl-CoA
C00479 Propanal

C00100 C00479

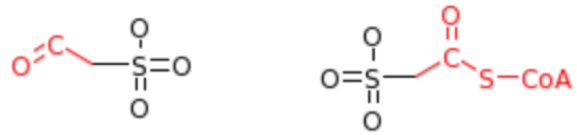
C00136 Butanoyl-CoA
C01412 Butanal

C00154 Palmitoyl-CoA
C00517 Hexadecanal



C00154 C00517

C00593 Sulfoacetaldehyde
C19685 Sulfoacetyl-CoA

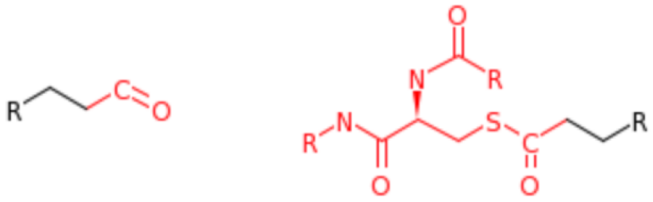


C00593 C19685

C00609 Long-chain aldehyde
C02843 Long-chain acyl-CoA

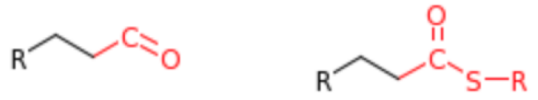
C00609 C02843

C00609 Long-chain aldehyde
C03371 [Protein]-S-(long-chain-acyl)-L-cysteine



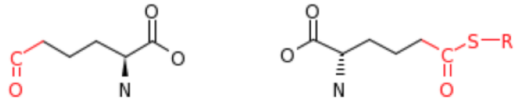
C00609 C03371

C00609 Long-chain aldehyde
C20683 Long-chain acyl-[acyl-carrier protein]



C00609 C20683

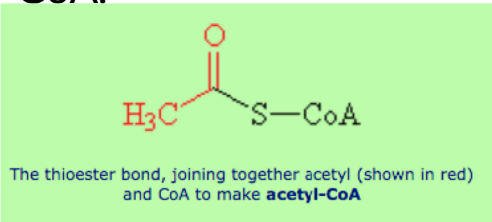
C04076 L-2-Aminoadipate 6-semialdehyde
C05535 alpha-Aminoadipoyl-S-acyl enzyme



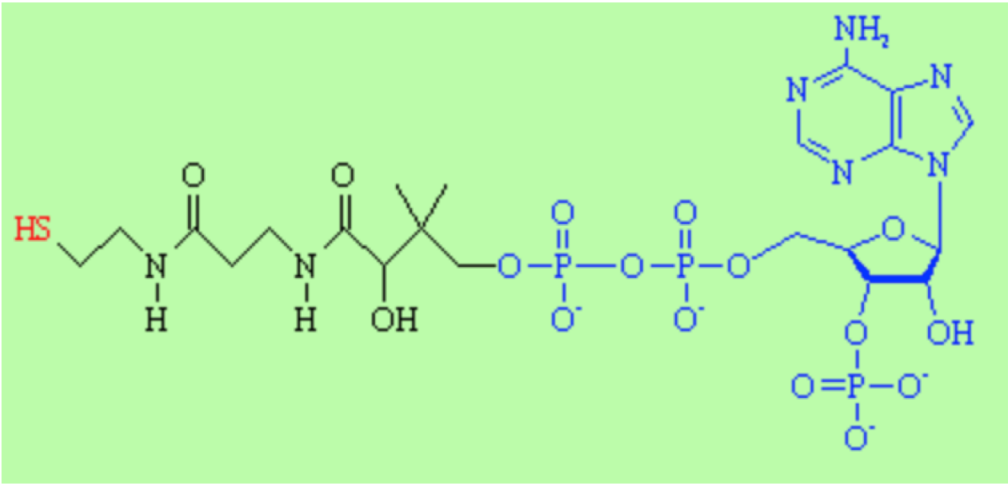
C04076 C05535

CoA – Acetyl Coenzyme A

- CoA is a coenzyme (a molecule that helps an enzyme)
- CoA is composed of two main parts, a long protein-like chain joined to adenosine diphosphate, ADP (used for energy storage (blue))
- The important part of the molecule is at the end of the protein chain, which terminates in a sulph-hydryl (-SH) group (red). This group is highly reactive.
- The most important acid is acetic acid, and when it is joined to CoA, the resulting compound is known as acetyl-CoA.



The thioester bond, joining together acetyl (shown in red) and CoA to make **acetyl-CoA**



http://www.chm.bris.ac.uk/motm/acetylcoa/acoah.htm

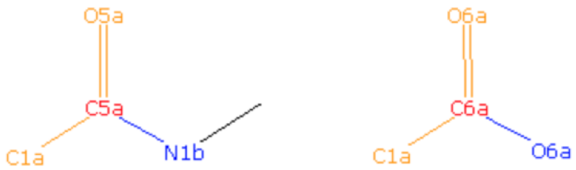
RClass 00300

- Transforms C00033 to many other compounds

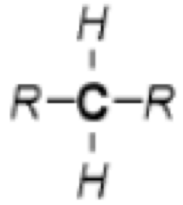
www.kegg.jp/dbget-bin/www_bget?rc:RC00300

RCLASS: RC00300

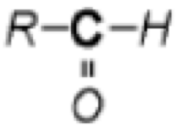
Help

Entry	RC00300 RClass
Definition	C5a-C6a:N1b+*-*+O6a:C1a+O5a-C1a+O6a
	
Reactant pair	C00033_C00140 C00033_C00357 C00033_C00437 C00033_C00461 C00033_C01029 C00033_C01042 C00033_C01073 C00033_C01215 C00033_C01288 C00033_C02714 C00033_C02727 C00033_C02946 C00033_C02998 C00033_C03087 C00033_C03357 C00033_C04390 C00033_C04394 C00033_C04690 C00033_C04738 C00033_C05548 C00033_C05727 C00033_C06376 C00033_C06746 C00033_C07032 C00033_C15532 C00033_C17582 C00033_C17587 C00033_C17951 C00033_C19784 C00033_C19929 » show all Path search
Related class	DB search
Reaction	R00458 R00488 R00669 R00909 R01156 R01200 R01649 R01987 R02059 R02276 R02333 R02733 R03482 R04056 R04174 R04397 R04587 R04727 R05168 R05677 R07300 R07301 R08876 R08895 R08901 R09107 R09651 R09721 R09801 R10553 R11283

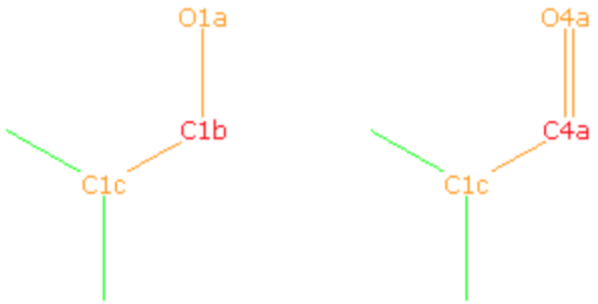
RC00099
Many different RPAIRS in the same RCLASS



C1b (20193)



C4a (350)

Entry	RC00099 RClass
Definition	C1b-C4a:*-*:C1c+O1a-C1c+O4a
	
Reactant pair	C00065_C11822 C00116_C00577 C00116_C02426 C00204_C04471 C00258_C01146 C00349_C01188 C00424_C00583 C00424_C02917 C00583_C05999 C00717_C01370 C00860_C01929 C00937_C02912 C01040_C02670 C01115_C06430 C01301_C05446 C01488_C01569 C05444_C05445 C05576_C05577 C06001_C06002 C16159_C16390 C16586_C16587 C19589_C19591 C19589_C19592 C19590_C19591 C19590_C19592 C20143_C21304 C20797_C20798 C21303_C21305 Path search

Summary for each RClass

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	...	...	..	...
...	...	...	...	...

- What processing of data is needed to generate this summary?

Handing in homework and such

- Use Gradscope
- Please provide all files that are needed to run your code
- README on how to run your code, and on the environment
  - If you are using anaconda, you can do that as follows:

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