Representing chemical data from Reaction Mechanism Generator (RMG) in a relational database

Belinda Slakman

April 30, 2015

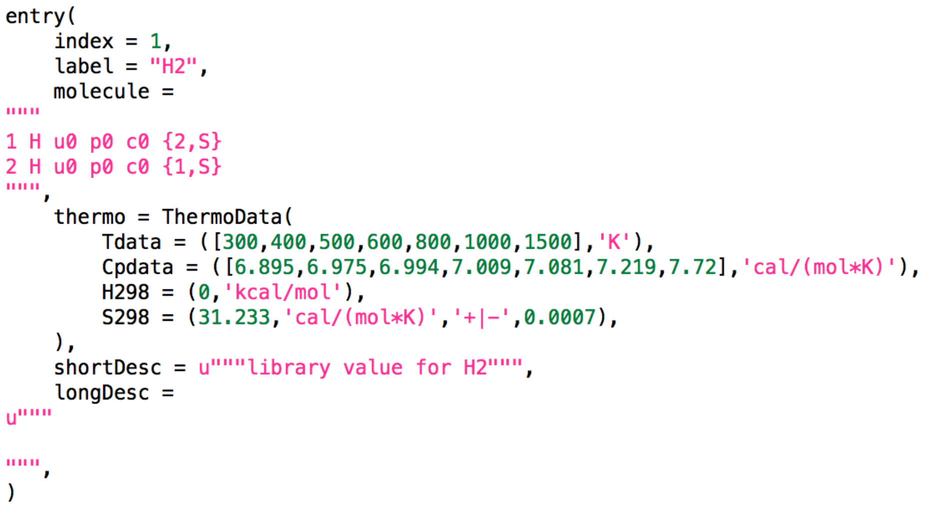
1 Introduction

Reaction Mechanism Generator (RMG) is an open-source software used to build detailed chemical kinetic models, given starting chemical species and process conditions such as initial concentrations, temperature, pressure, and solvent [1]. RMG, written in Python, is generally used to build models for hydrocarbon combustion, but recent additions allow some functionality for oxygenated biofuels, liquid-phase reactions, and silicon hydrides. For generating these models, a large number of thermodynamic and kinetic parameters must be known or estimated from known values. The parameters that various users and developers have added to RMG are stored in a database.

Currently, RMG's database is stored in a series of Python files. Each entry in the database is associated with a particular chemical species or a reaction. The structure is similar to a key-value structure where each entry has a label and a set of values associated. Figure 1 shows an example of an entry in a thermodynamic library.

The problem with the RMG database's current structure is that data is looked up constantly on-the-fly during the course of a simulation. When we encounter a new species, we must look up its thermodynamic and solvation data in separate libraries. Furthermore, if multiple thermodynamic libraries were specified in the input file, we will only use the value from the first library listed, even if the species exists in other libraries with different thermodynamic values. The index associated for each entry in a library is also relatively meaningless. Another issue is that chemical species cannot be specified uniquely; they are specified with a non-canonical label and an adjacency list. Graph matching algorithms are used to compare the adjacency lists. It would be desirable to both reduce the number of lookups and increase the speed of the lookups by simplifying the database currently used.

A relational database structure could solve some of these issues. Relational data lookup is fast, and the data we are working with is mainly in text form. A relational database can also better represent relationships between data; molecules may appear in multiple reactions in kinetics libraries, as well as be associated with thermodynamic data in several libraries, and have associated



```
entry(
    index = 1,
    label = "H2",
    molecule = 
"""
1 H u0 p0 c0 {2,S}
2 H u0 p0 c0 {1,S}
""",
    thermo = ThermoData(
        Tdata = ([300,400,500,600,800,1000,1500],'K'),
        Cpdata = ([6.895,6.975,6.994,7.009,7.081,7.219,7.72],'cal/(mol*K)'),
        H298 = (0,'kcal/mol'),
        S298 = (31.233,'cal/(mol*K)','+|-',0.0007),
    ),
    shortDesc = u"""library value for H2""",
    longDesc = 
u"""

""",
)
```

Figure 1: Example of an entry from a thermodynamic library in RMG-database. The entry contains an index, label, adjacency list, thermodynamic data, and comments.

solvation data. Furthermore, users who are not performing an RMG simulation, but have general questions about the data available or about some reactions or chemical species, can easily and quickly perform queries on this relational database.

2 Description of project aims

The aims of the term project will be described below.

2.1 Demonstrate web scraping concepts to retrieve parameters from the RMG database

In addition to the Python files, the RMG-database is also available on the web at both `rmg.coe.neu.edu/database` and `rmg.mit.edu/database`. To demonstrate the web scraping methods from this course with the XML and RCurl packages in R, thermodynamic, kinetic and solvation data will be scraped from various libraries and sorted into data frames.

2.2 Restructure parts of the RMG database into a relational database

By reorganizing data and using the SQLite package in R, the data collected in the first aim will be organized into a relational database that satisfies third normal form.

2.3 Perform queries of data of interest to someone working with RMG or with chemical data

Ability to query the database, again using the SQLite package in R, will be demonstrated. From the result of these queries we can understand the strengths and weaknesses of using a relational database structure to represent RMG's database.

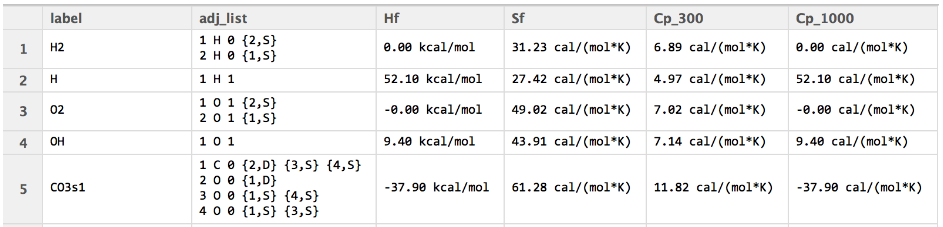
3 Web Scraping

Methods written to scrape thermodynamic, kinetic and solvation data are contained in the attached file `RMGWebScraping.R`. This file depends on the packages RCurl and XML. It contains three methods: `ScrapeRMGThermo()`, `ScrapeRMGKineticsFromLibrary()`, and `ScrapeRMGSolvation()`, each returning a data frame of the quantities of interests. These will be described in detail below.

3.1 Thermodynamics

The thermodynamic data can come from two different source types, libraries or groups, with each type having different libraries or sets of groups. For example, thermodynamic libraries include sets of parameters determined from quantum mechanics calculations, such as "DFT_QCI_Thermo", or from other groups' work, including "GRI-Mech3.0". Each library be associated with a different research group or set of chemical species. Thermodynamic group values belong to molecular structure groups that make up a chemical species, such as radicals or rings. Despite differences between full molecules and groups, the webpages are structured mostly the same, so we have the option of scraping differently. However, the groups are listed in a tree based on molecular structure allowing for easier lookup; we cannot replicate this tree in a relational database.

The RCurl package is used to retrieve URLs and the XML package is used to process the webpage and retrieve nodes (using xslt language). The number of species is found by retrieving the last node on the library or group page; this is useful because we can then pre-allocate vectors of values, saving on computational time when looping through each molecule or group. The information retrieved and parsed includes the molecule label (name or chemical formula), its adjacency list, the enthalpy of formation (Hf), entropy of formation (Sf), and heat capacity at two different temperatures (Cp_300 and Cp_1000). Hf, Sf, Cp_300, and Cp_1000 are only retrieved if the thermo is in the format "Group additivity". One could imagine having other scraping functions for thermo given in different forms, such as NASA polynomials, but for simplicity only one is included here. Rows of all 'NA' values, meaning no webpage existed for a particular index so it was skipped, are removed before returning the data frame. An example of a data frame of thermodynamic library data is given in Figure 2.



	label	adj_list	Hf	Sf	Cp_300	Cp_1000

1	H2	1 H 0 {2,S}
		2 H 0 {1,S}
			0.00 kcal/mol	31.23 cal/(mol*K)	6.89 cal/(mol*K)	0.00 cal/(mol*K)

2	H	1 H 1	52.10 kcal/mol	27.42 cal/(mol*K)	4.97 cal/(mol*K)	52.10 cal/(mol*K)

3	O2	1 O 1 {2,S}
		2 O 1 {1,S}
			-0.00 kcal/mol	49.02 cal/(mol*K)	7.02 cal/(mol*K)	-0.00 cal/(mol*K)

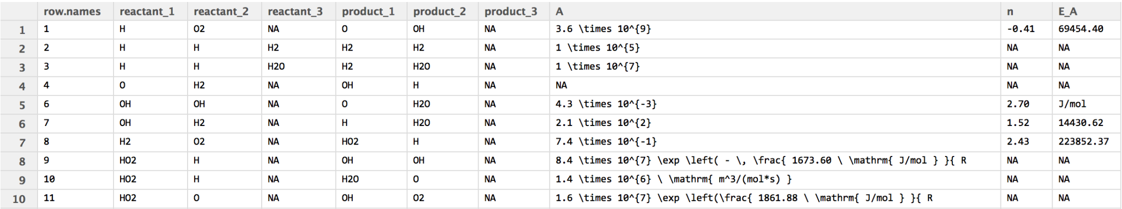
4	OH	1 O 1	9.40 kcal/mol	43.91 cal/(mol*K)	7.14 cal/(mol*K)	9.40 cal/(mol*K)

5	CO3s1	1 C 0 {2,D} {3,S} {4,S}
		2 O 0 {1,D}
		3 O 0 {1,S} {4,S}
		4 O 0 {1,S} {3,S}
			-37.90 kcal/mol	61.28 cal/(mol*K)	11.82 cal/(mol*K)	-37.90 cal/(mol*K)

Figure 2: Result of running the scraping method `ScrapeRMGThermo('libraries', 'primaryThermoLibrary')`. Only the first five entries are shown.

3.2 Kinetics

The web scraping protocol for RMG kinetics is similar to that for thermodynamics, but we do not include an option for scraping the kinetics groups since the format is very different than that of the libraries. The values scraped are the reactants (by name or chemical formula, up to 3), products (up to 3), and Arrhenius kinetics parameters including the pre-exponential factor (A), temperature exponent (n), and activation energy (E_A). If kinetics is not Arrhenius, we skip the scraping for that reaction; again, one could add different protocols for scraping kinetics of different formats.



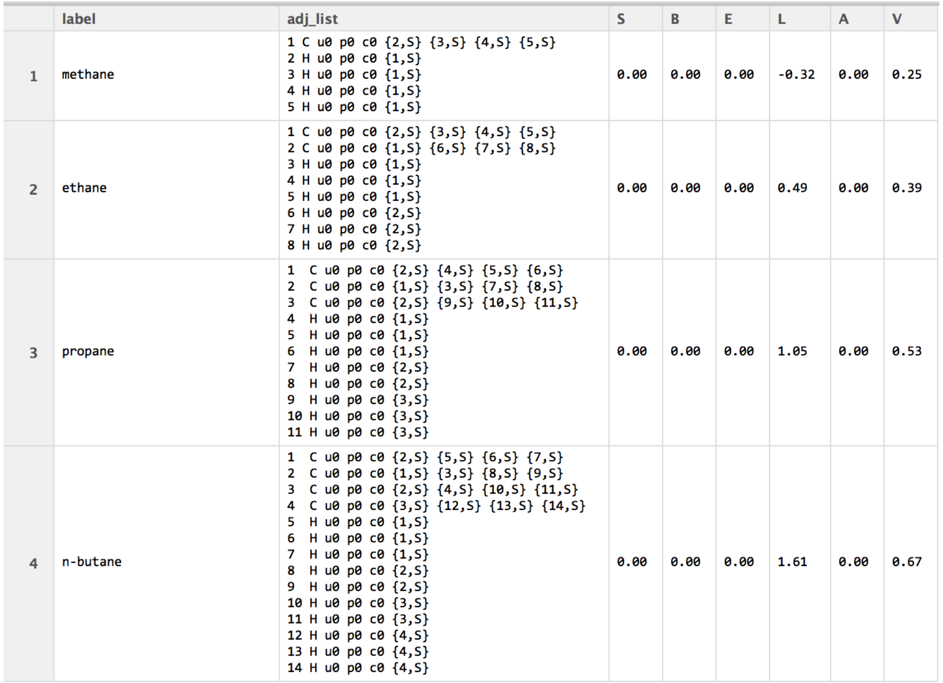
	row.names	reactant_1	reactant_2	reactant_3	product_1	product_2	product_3	A	n	E_A
1	1	H	O2	NA	O	OH	NA	3.6 \times 10^{9}	-0.41	69454.40
2	2	H	H	H2	H2	H2	NA	1 \times 10^{5}	NA	NA
3	3	H	H	H2O	H2	H2O	NA	1 \times 10^{7}	NA	NA
4	4	O	H2	NA	OH	H	NA	NA	NA	NA
5	6	OH	OH	NA	O	H2O	NA	4.3 \times 10^{-3}	2.70	J/mol
6	7	OH	H2	NA	H	H2O	NA	2.1 \times 10^{2}	1.52	14430.62
7	8	H2	O2	NA	HO2	H	NA	7.4 \times 10^{-1}	2.43	223852.37
8	9	HO2	H	NA	OH	OH	NA	8.4 \times 10^{7} \exp \left(- \, \frac{ 1673.60 \ \mathrm{ J/mol } }{ R	NA	NA
9	10	HO2	H	NA	H2O	O	NA	1.4 \times 10^{6} \ \mathrm{ m^3/(mol*s) }	NA	NA
10	11	HO2	O	NA	OH	O2	NA	1.6 \times 10^{7} \exp \left(\frac{ 1861.88 \ \mathrm{ J/mol } }{ R	NA	NA

Figure 3: Result of running the scraping method `ScrapeRMGKineticsFromLibrary('GRI-Mech3.0')`; only the first five entries are shown. Note that the cleaning of the data did not work for all of the kinetic parameters.

The data frame created from the web scraping is shown in Figure 3. Note that the text processing fails on separating some of the Arrhenius parameters into A, n and E_A; using a different web scraping tool could make the data cleaning easier.

3.3 Solvation

Scraping the solvation data is similar to thermodynamics and kinetics, but it is easier because there is only one solute library available in RMG so it is scraped by default. The solute parameters (S, B, E, L, A and V) are coefficients in a linear solvation energy relationship that relates gas and solution phase thermodynamics for a chemical species. A portion of the data frame resulting from scraping the solvation library is illustrated in Figure 4.



	label	adj_list	S	B	E	L	A	V

1	methane	1 C u0 p0 c0 {2,S} {3,S} {4,S} {5,S}
		2 H u0 p0 c0 {1,S}
		3 H u0 p0 c0 {1,S}
		4 H u0 p0 c0 {1,S}
		5 H u0 p0 c0 {1,S}
			0.00	0.00	0.00	-0.32	0.00	0.25

2	ethane	1 C u0 p0 c0 {2,S} {3,S} {4,S} {5,S}
		2 C u0 p0 c0 {1,S} {6,S} {7,S} {8,S}
		3 H u0 p0 c0 {1,S}
		4 H u0 p0 c0 {1,S}
		5 H u0 p0 c0 {1,S}
		6 H u0 p0 c0 {2,S}
		7 H u0 p0 c0 {2,S}
		8 H u0 p0 c0 {2,S}
			0.00	0.00	0.00	0.49	0.00	0.39

3	propane	1 C u0 p0 c0 {2,S} {4,S} {5,S} {6,S}
		2 C u0 p0 c0 {1,S} {3,S} {7,S} {8,S}
		3 C u0 p0 c0 {2,S} {9,S} {10,S} {11,S}
		4 H u0 p0 c0 {1,S}
		5 H u0 p0 c0 {1,S}
		6 H u0 p0 c0 {1,S}
		7 H u0 p0 c0 {2,S}
		8 H u0 p0 c0 {2,S}
		9 H u0 p0 c0 {3,S}
		10 H u0 p0 c0 {3,S}
		11 H u0 p0 c0 {3,S}
			0.00	0.00	0.00	1.05	0.00	0.53

4	n-butane	1 C u0 p0 c0 {2,S} {5,S} {6,S} {7,S}
		2 C u0 p0 c0 {1,S} {3,S} {8,S} {9,S}
		3 C u0 p0 c0 {2,S} {4,S} {10,S} {11,S}
		4 C u0 p0 c0 {3,S} {12,S} {13,S} {14,S}
		5 H u0 p0 c0 {1,S}
		6 H u0 p0 c0 {1,S}
		7 H u0 p0 c0 {1,S}
		8 H u0 p0 c0 {2,S}
		9 H u0 p0 c0 {2,S}
		10 H u0 p0 c0 {3,S}
		11 H u0 p0 c0 {3,S}
		12 H u0 p0 c0 {4,S}
		13 H u0 p0 c0 {4,S}
		14 H u0 p0 c0 {4,S}
			0.00	0.00	0.00	1.61	0.00	0.67

Figure 4: Result of running the scraping method `ScrapeRMGSolvation()`.

4 Transformation into a relational database

Data manipulations were performed in R to translate the scraped data into three tables for a relational database. The manipulations, and the following queries were done in the attached file `RMGDatabase.R`. Examples of three thermodynamic libraries, three kinetic libraries, and the solvation library were scraped for data. An explanation of each table, and how the database satisfies third normal form, is below.

4.1 Thermodynamic data table

The primary keys in the thermodynamic table are the species identifier, *label*, and the thermodynamic library it belongs to, *thermoLibraryName*. Together, these keys uniquely specify a single thermodynamic entry, as the same molecule can only appear once in a thermo library (although it can appear in several libraries). The table also contains thermodynamic parameters.

4.2 Molecule and solvation data table

The second table contains the each molecule's label, adjacency list and its solvation data. The primary key is the *label*. The adjacency list and the solvation data should be unique to a molecule, since there is only solute library and the adjacency list for a given molecule should give the same chemical graph. Note that this does NOT mean that the adjacency list string is unique for a given molecule, because atoms can be specified in any order. This is why we select unique rows from the resulting table, since we can have the same molecule multiple times with different adjacency lists (that mean the same thing, so it doesn't matter what row we keep).

4.3 Kinetic data

The table containing the kinetic data contains an index, the name of the kinetics library, the reactants and products, and the Arrhenius kinetic data. The *index* and the *kineticsLibraryName* uniquely specify the reaction and are used as the primary key. This allows a reaction with the same reactants and products, but different kinetics, to be listed more than once.

4.4 Satisfying 3NF

The database satisfies 1NF since each has a primary key, and no table has multiple values for a single column. Also, each column's value depends on the primary key of the table. This is why adjacency list and solute data were placed in the same table, since neither of these depends on the thermodynamic or kinetic library the molecule is in.

2NF applies to the thermodynamics and kinetics tables, which have multiple columns as primary keys. The values in the other columns in each table definitely depend on the thermodynamic library or kinetic library, respectively; a label or an index was not sufficient since a single molecule or reaction can appear in multiple libraries.

The database also satisfies 3NF because no columns are dependent on anything but the primary keys. One could argue that some reaction rates are known so well that the Arrhenius parameters are dependent on the reactants and products themselves and not the index and reaction library; however, inconsistencies still occur between different experimental and calculation methods (and no reaction rate is known that accurately!)

5 Querying the database

Chemical data queries are performed in the file `RMGDatabase.R` via SQL SELECT statements. A summary of the results follows:

- OH appears in both the primaryThermoLibrary and DFT_QCI_Thermo libraries.
- The chemical species that appear in more than one library are H2, O2, and OH. These queries are useful, because we can compare the thermodynamic parameters in each and if there is a large discrepancy, it motivates further experimentation or calculation.
- In the three libraries, H2 appears as a reactant or product 94 times. Knowing how many times a species appears also can motivate further research.

- 62 reactions in GRI-Mech3.0, 162 in Glarborg/C3, and 23 in Sulfur/DMS have kinetics in Arrhenius format. Note that these numbers may be incorrect due to faulty scraping.
- None of the molecules in the primaryThermoLibrary contain solvation data. This is not actually true, but why this result occurs will be discussed.

6 Discussion

While web scraping using XML in R and subsequent cleaning of data could be done reasonably well for thermodynamic and solvation data, we had an issue with the Arrhenius parameters. To this end, one of the web scraping tools in the class might be better suited to capture these parameters; though they are less customizable, they require less user input. Kimono was the option I found easiest to use during the course exploration of tools.

For both scraping and manipulating the data into a relational database, we found that libraries of data represented by entire molecules were easier to deal with than group data, in which values are defined by molecular structure groups. Because groups are represented by hierarchical trees, it would be beneficial to represent this tree in a graph database where relationships, such as parent and child, are known between nodes.

Querying of the database succeeded for most of the queries; however, it failed when we looked for solvation data for molecules in a particular thermodynamic library. This failure is due to a flaw in the keys used to represent molecules. We inherently assume that the label representation of a molecule is the same across all libraries, but the way RMG actually does these comparisons is through graph matching algorithms of the adjacency lists. However, the solvation library was created with name labels (e.g. methane) and the thermodynamic and kinetic libraries mainly use chemical formula labels (e.g. CH4), which is why our query failed. Graph matching introduces a speed limitation in RMG, because if we could instead do simple string comparison, all queries would be much quicker (regardless of the database structure used). However, although strings (such as SMILES [2]) exist to identify molecules, they are non-canonical, meaning they do not uniquely specify the chemical species. Recently, Burgess et al. have suggested a method for uniquely identifying molecules [3].

To summarize, relational databases are a quick and dirty way to store and retrieve data for libraries of chemical data. However, full functionality for the RMG database requires some additional data structures to represent hierarchical trees, as well as a canonical way to represent chemical species as strings.

References

[1] William H. Green, Joshua W. Allen, Pierre Bhoorasingh, Beat A. Buesser, Robert W. Ashcraft, Gregory J. Beran, Caleb A. Class, Connie Gao, C. Franklin Goldsmith, Michael R. Harper, Amrit Jalan, Fariba Seyedzadeh Khanshan, Gregory R. Magoon, David M. Matheu, Shamel S. Merchant, Jeffrey D. Mo, Sarah Petway, Sumathy Raman, Sandeep Sharma, Belinda Slakman, Jing Song, Kevin M. Van Geem, John Wen, Richard H. West, Andrew Wong, Hsi-Wu Wong, Paul E. Yelvington, Nathan Yee, and Joanna Yu. RMG — Reaction Mechanism Generator-Python. *rmg.mit.edu*, pages RMG — Reaction Mechanism Generator, 2013.

[2] David Weininger. SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. *J. Chem. Inf. Comput. Sci.*, 28:31–36, 1988. ISSN 15499596. doi: 10.1021/ci00057a005.

[3] Donald R. Burgess, Jeffrey a. Manion, and Carrigan J. Hayes. Data Formats for Elementary Gas Phase Kinetics, Part 1: Unique Representations of Species at the Molecular Level. *Int. J. Chem. Kinet.*, 46:640–650, 2014. ISSN 05388066. doi: 10.1002/kin.20875. URL `http://doi.wiley.com/10.1002/kin.20875`.