Reaction Mechanism Generator v3.0: Advances in Automatic Mechanism Generation

Mengjie Liu,[†] Alon Grinberg Dana,[†,‡] Matthew S. Johnson,[†] Mark J. Goldman,[†] Agnes Jocher,[†] A. Mark Payne,[†] Colin A. Grambow,[†] Kehang Han,[†] Nathan W. Yee,[†] Emily J. Mazeau,[¶] Katrin Blondal,[§] Richard H. West,[¶] C. Franklin Goldsmith,[§] and William H. Green[*,†]

†*Department of Chemical Engineering, Massachusetts Institute of Technology, Cambridge, MA 02139, United States*

‡*Wolfson Department of Chemical Engineering, Technion – Israel Institute of Technology, Haifa 3200003, Israel*

¶*Department of Chemical Engineering, Northeastern University, Boston, MA 02115, United States*

§*School of Engineering, Brown University, Providence, RI 02912, United States*

E-mail: whgreen@mit.edu

Abstract

In chemical kinetics research, kinetic models containing hundreds of species and tens of thousands of elementary reactions are commonly used to understand and predict the behavior of reactive chemical systems. Reaction Mechanism Generator (RMG) is a software suite developed to automatically generate such models by incorporating and extrapolating from a database of known thermochemical and kinetic parameters.

Here, we present the recent version 3 release of RMG and highlight improvements since the previously published description of RMG v1.0. One important change is that RMG v3.0 is now Python 3 compatible, which supports the most up-to-date versions of cheminformatics and machine learning packages that RMG depends on. Additionally, RMG can now generate heterogeneous catalysis models, in addition to the previously available gas- and liquid-phase capabilities. For model analysis, new methods for local and global uncertainty analysis have been implemented to supplement first-order sensitivity analysis. The RMG database of thermochemical and kinetic parameters has been significantly expanded to cover more types of chemistry. The present release also includes parallelization for reaction generation and on-the-fly quantum calculations, and a new molecule isomorphism approach to improve computational performance. Overall, RMG v3.0 includes many changes which improve the accuracy of the generated chemical mechanisms and allow for exploration of a wider range of chemical systems.

Introduction

Detailed chemical kinetic modeling continues to gain interest as an approach to study reactive chemical systems, ranging in application from combustion and pyrolysis of fuels to degradation of active pharmaceutical ingredients. This growth can be attributed to a combination of demand for studying increasingly complex chemistries and supply of computational power and quantum chemistry capabilities. By taking advantage of these computational resources, automatic mechanism generation tools[1–6] are able to systematically enumerate and evaluate potential chemical pathways, reducing the chance of human error. This is largely a data-driven task, requiring good estimation algorithms for thermochemical and rate parameters, which in turn rely on accurate training data from experiments or quantum chemistry calculations.

The Reaction Mechanism Generator (RMG) project has been in development for over a decade, with the current Python version having begun development in 2008. RMG v1.0

was described in 2016.[7] Here, we are excited to present RMG v3.0, which brings many new features including Python 3 compatibility, heterogeneous catalysis modeling, and new parameter estimation algorithms. With these and other improvements, the codebase has doubled to over 120,000 lines of Python code. Many developments have been focused on improving nitrogen, sulfur, and aromatic chemistry to better model combustion emissions and refining processes. RMG has recently been used successfully to model ethylamine pyrolysis,[8] di-tert-butyl sulfide pyrolysis,[9] hexylbenzene pyrolysis,[10] effect of substituted phenols on ignition delay,[11] PAH formation in methane oxidation,[12] and catalytic combustion of methane.[13]

The structure and concept behind RMG has been described previously,[7] so only a brief overview will be given here. RMG is a tool for automatically constructing detailed chemical mechanisms which is largely comprised of three components:

1. a cheminformatics framework for representing molecules, reactions, and various data classes for thermochemistry and kinetics

2. a database and parameter estimation framework for predicting thermochemistry and kinetics parameters

3. a mechanism construction framework, primarily using a flux-based species selection algorithm, including functionality for automatic construction of pressure-dependent networks.

The latest release of RMG includes updates across all three components to expand modeling capabilities and improve accuracy, robustness, and performance.

RMG uses a core/edge reaction model during mechanism generation, where the core contains species and reactions which have already been identified as being important, and the edge contains species and reactions which are under consideration. To reduce the model truncation error,[14,15] in each iteration, RMG identifies one or more species to move from the edge to the core based on the species' total formation rate in a homogeneous batch reactor

simulation. It then generates new reactions between the newly added species and other species in the core. The model is considered converged when no edge species exceeds the user-specified tolerance for selection.

New features in RMG

Python 3 compatibility

The official end-of-life for Python 2, January 1, 2020, motivated many software projects to transition to Python 3, including leading cheminformatics packages like RDKit and Cantera. In order to stay up-to-date with these software, it was essential for RMG to upgrade to Python 3 as well.

The transition for RMG included many steps. The first step was ensuring that Python 3 versions of all of our dependencies were available. This was straightforward for widely-used packages since all of them already supported Python 3. However, some packages developed specifically for RMG also had to be updated with Python 3 support, namely PyDAS and PyDQED.[16,17] The second step of modifying RMG for Python 3 compatibility was facilitated by automatic tools like python-future, although substantial manual intervention was still required. In the final step, we used this opportunity to standardize function names throughout our API to comply with PEP-8 recommendations, effectively the official Python style guide. In total, transition tasks took approximately 500 developer hours to complete.

With the v3.0 release, RMG is now fully compatible with Python 3.7. The Python 2 version of RMG will no longer be actively supported, although a legacy version will be made available for users.

Heterogeneous catalysis

RMG v3.0 also introduces support for generating heterogeneous catalysis models, which was previously developed independently as the RMG-Cat project.[18] This feature involved

additions to all aspects of the model generation process.

Molecule representations have been extended to include catalyst sites, which are represented as a generic "X" element. New bond types have been implemented to represent the metal-adsorbate bond, including van der Waals bonds (internally represented with a bond order of 0) and quadruple bonds (e.g., for adsorption of a carbon atom). These extend the existing single, double, triple, and benzene bond orders.

Thermochemistry estimation has been expanded to estimate parameters for surface species by applying adsorption corrections. For a given surface species, the metal is first removed to obtain an estimate for the gas-phase species using existing methods (e.g., group additivity or libraries), then an adsorption correction is determined from a group additivity tree and added to the gas-phase value. Thermochemistry libraries are also supported for surface species. The RMG database currently contains a thermochemistry library with 21 adsorbates on nickel and a more recent library that has 69 H/C/O/N-containing adsorbates on platinum. By default, RMG uses binding energies for Pt(111), but energies for an arbitrary catalyst can be specified in the input file (Figure 1). Adsorption corrections are then scaled appropriately based on the specified binding energies.

```
catalystProperties(
    bindingEnergies={
        'H': (-2.479, 'eV/molecule'),
        'O': (-3.586, 'eV/molecule'),
        'C': (-6.750, 'eV/molecule'),
        'N': (-4.352, 'eV/molecule'),
    },
    surfaceSiteDensity=(2.72e-9, 'mol/cm^2'),
)
```

Figure 1: Example input file block for specifying catalyst properties.

Kinetics estimation has been expanded with new families (detailed in the Kinetics section) for estimating various types of surface reactions, such as adsorption and dissociation. To support these surface reactions, new data classes have also been added for surface rate con-

stants (`SurfaceArrhenius` and `SurfaceArrheniusBEP` for Bronsted-Evans-Polanyi relationships) and sticking coefficients (`StickingCoefficient` and `StickingCoefficientBEP`).

Surface simulations require use of the new `SurfaceReactor` class. This module performs the reactor simulations necessary for the flux-based algorithm for model growth. It is modeled as a zero-dimensional, isothermal, isochoric batch reactor which tracks surface coverage in addition to gas-phase mole fractions. User specification of surface area to volume ratio and surface site density are required. For surface mechanism generation jobs, RMG will output separate gas- and surface-phase Chemkin mechanism files along with a single Cantera mechanism file.

A recent case study in methane catalytic combustion on platinum[13] demonstrates the heterogeneous catalysis functionality. It addresses extensive updates to the original release of RMG-Cat.[18] Among those is the new platinum thermochemistry database which is larger and more accurate than the original nickel database and has the advantage of including nitrogen-containing adsorbates. Another important new feature is the ability to explore heterogeneous and gas-phase reactions simultaneously, as with that the resulting microkinetic models can provide an implication of when catalytic surfaces lead to radical chemistry in the gas phase.

Uncertainty analysis

Beyond generating chemical mechanisms, RMG also provides features for model analysis. Previously, local first-order sensitivity analysis was available to calculate sensitivities of species concentrations to thermochemistry and rate constants. New methods for both local and global uncertainty analysis have been implemented in RMG.[19] Local uncertainty analysis builds on those first-order sensitivity by incorporating estimated uncertainties for thermochemical and rate parameters to obtain uncertainties for species concentrations. Global uncertainty analysis uses the MIT Uncertainty Quantification Library (MUQ 2)[20] to construct polynomial chaos expansions (PCEs) based on reactor simulations at random points

within the uncertainty space of the input thermochemical and rate parameters. Reactor simulations are performed using Cantera.[21] A key feature of the RMG uncertainty module is the ability to track correlated uncertainties in model input parameters, such as correlations arising from group additivity estimates for thermochemistry and rate rule estimates for rate coefficients. This can have significant effects on uncertainty propagation and the resulting uncertainties on output parameters.

Uncertainty analysis can be requested via the RMG input file, which will lead to it being performed upon completion of model generation. Using uncertainty analysis does require that sensitivity analysis settings also be provided, since sensitivity analysis is required part of local uncertainty analysis. Local uncertainty analysis is also used to determine the parameters to vary for global uncertainty analysis, in order to minimize computational cost. For global analysis, PCE fitting can be controlled by specifying either a maximum runtime, error tolerance, or maximum number of model evaluations. These methods can also be applied to already-generated models via standalone scripts and interactive Jupyter notebooks, with the limitation that the same RMG version must be used for both model generation and analysis. These new tools can provide insights beyond first-order sensitivity analysis to aid in the model development process.

```
uncertainty(
    localAnalysis=True,
    globalAnalysis=True,
    uncorrelated=True,
    correlated=True,
    localNumber=10,
    globalNumber=5,
    pceRunTime=1800,
    pceErrorTol=None,
    pceMaxEvals=None,
)
```

Figure 2: Example input file block for requesting uncertainty analysis.

Ranged reactors

In RMG, the reaction conditions of interest (i.e., temperature (T), pressure (P), initial composition (X_0)) are provided by defining reactors in the input file. RMG supports three reactor types for mechanism generation, distinguished by the phases involved: `SimpleReactor` for gas phase, `LiquidReactor` for liquid phase, and the new `SurfaceReactor`.

Because RMG uses a flux-based algorithm for identifying important species and reactions, the reactor conditions used to generate a model directly affect the conditions at which the model is applicable. Previously, the recommended approach for building a model applicable at a range of conditions was to define multiple reactors spanning the space of conditions of interest. For example, if the goal was to develop a model valid for temperatures from 1000 K to 2000 K and pressures from 1 bar to 10 bar, the user may need to define a dozen reactors with all combinations of $T = \{1000, 1200, 1400, 1600, 1800, 2000\}$ K and $P = \{1, 10\}$ bar. This can be bothersome to the user, and risks missing important chemistry which may occur in between the chosen points.

Ranged reactors are a new feature in RMG v3.0 to simplify the task of specifying a range of initial conditions. With the new feature, ranges for $T, P,$ and X_0 can be directly specified for a single reactor block. Internally, RMG will automatically select points within the space of conditions for each iteration, using a weighted stochastic grid sampling algorithm. On each iteration, a coarse grid with 20 points in each dimension is constructed, and the desirability of each point is evaluated based on the number of iterations since it was last chosen. The desirability values are normalized to form probabilities, and a random point is chosen using those probabilities. The algorithm then takes a random step from the chosen point, with a maximum distance of $\sqrt{2}/2$ times the distance between grid points. That point is then used for the simulation. The algorithm continues iterating through the grid points considering the probabilities described above. A simplified example of the algorithm for two dimensions is shown in Figure 3.

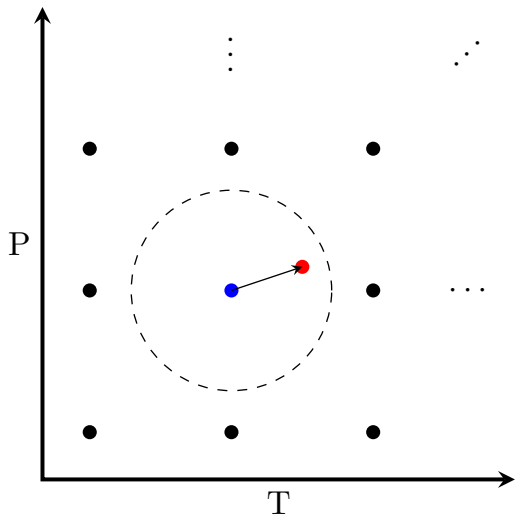


Figure 3: Schematic representation of how RMG selects conditions for a given simulation when using ranged reactors. The blue point indicates the initial point chosen from the coarse grid. A random step is then taken within the bounds of the dotted line, which results in the final set of conditions represented by the red point.

Isotopic mechanisms

RMG can now generate isotopically labeled reaction mechanisms via a post-processing algorithm.[22] After a normal RMG job is completed, the isotopes module can generate all combinations of isotopically labeled species and reactions (Figure 4). To obtain consistent thermodynamics, RMG modifies species' entropy based on changes to molecular symmetry and modifies kinetic Arrhenius factors based on reaction path degeneracy. Classical, mass-dependent kinetic isotope effects (KIE) are also available.

One challenge with this approach is that isotopically-labeled mechanisms grow exponentially with the number of atoms that can be isotopically labeled. For example, a single asymmetric molecule with six carbons would be represented by 64 different species with various carbon atoms enriched. Despite the combinatorial complexity, this method is still very useful for generating detailed isotopic mechanisms, and has been shown to provide good agreement and insight into position-specific isotope analysis experiments.[22,23] Currently, the algorithm is limited to generation of isotopic mechanisms for ^{13}C, though the framework is easily extensible to other isotopes.

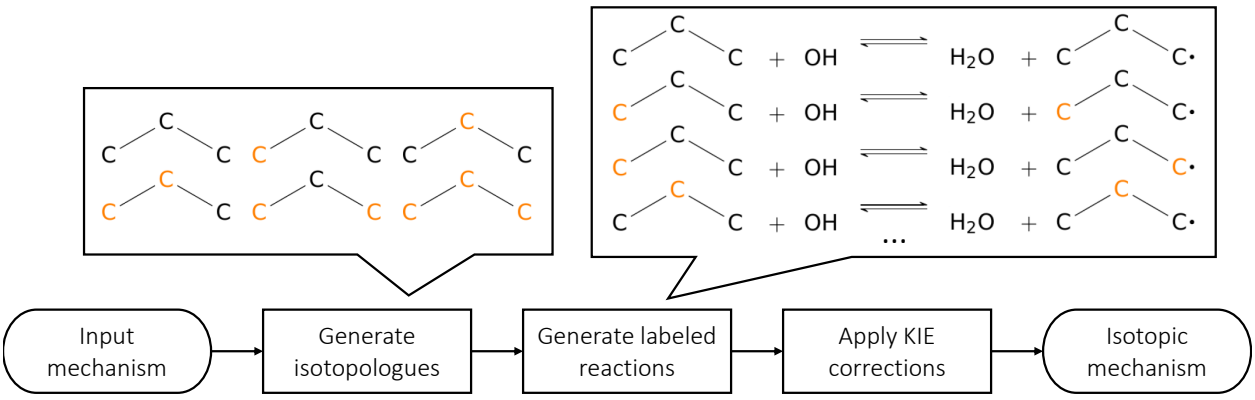


Figure 4: Algorithm for constructing isotopic reaction mechanisms. Orange indicates ^{13}C carbon atom. The left box shows the six ways propane can be labeled, and the right box shows four of the eight ways hydrogen abstraction from propane by OH can proceed.

Molecular representation

Atom types

Atom types in RMG are a set of atom descriptors that describe the local environment around an atom. They can accelerate graph isomorphism (by using specific types) and improve flexibility when defining reactions (by using more generic types). The set of available atom types has been revised and expanded to improve representation of heteroatoms. Particular focus has been placed on expanding atom type descriptors for the various bonding configurations of nitrogen and sulfur, for which the full list of updated atom types has been recently reported.[24] New carbon and oxygen atom types for representing formal charges and varying numbers of lone pairs have been added, along with additional halogen atom types. For surface chemistry, atom types representing generic surface sites have been added, along with quadruple bonds for carbon and silicon. A list of these new atom types is shown in Table 1.

Resonance structures

Resonance structures are an important aspect of molecule representation in RMG. Given that RMG uses localized representations of molecules (i.e., Lewis structures), it is important

Table 1: New atom types available in RMG v3.0

Atom Type	Description
New carbon atom types	
Ca	Carbon atom with two lone pairs
Csc	Carbon with all single bonds and formal charge of +1
Cdc	Carbon with one double bond and formal charge of +1
Cq	Carbon with quadruple bond (for surface adsorption)
C2s	Carbon with one lone pair and single bonds
C2sc	Carbon with one lone pair, single bonds, and formal charge of -1
C2d	Carbon with one lone pair and one double bond
C2dc	Carbon with one lone pair, one double bond, and formal charge of -1
C2tc	Carbon with one lone pair, one triple bond, and formal charge of -1
New oxygen atom types	
O0sc	Oxygen with three lone pairs, single bonds, and formal charge of -1
O2s	Oxygen with two lone pairs and single bonds
O2sc	Oxygen with two lone pairs, single bonds, and formal charge of +1
O2d	Oxygen with two lone pairs and one double bond
O4sc	Oxygen with one lone pair, single bonds, and formal charge of +1
O4dc	Oxygen with one lone pair, one double bond, and formal charge of +1
O4tc	Oxygen with one lone pair, one triple bond, and formal charge of +1
O4b	Oxygen with one lone pair and two benzene bonds
New halogen atom types	
F	Fluorine with any local bonding structure
F1s	Fluorine with three lone pairs and one single bond
Cl	Chlorine with any local bonding structure
Cl1s	Chlorine with three lone pairs and one single bond
I	Iodine with any local bonding structure
I1s	Iodine with three lone pairs and one single bond
New silicon atom types	
Siq	Silicon with quadruple bond (for surface adsorption)
New surface site types	
X	Generic surface site
Xv	Vacant surface site
Xo	Occupied surface site

that the algorithm can generate and identify the structures which are most representative of the true behavior of a molecule in terms of reactivity. Thus, significant improvements have been made to resonance structure generation algorithms, in particular for aromatic species and heteroatoms.[24,25] For aromatic species, RMG can now generate Clar structures[26,27] in replacement of Kekulé structures which are considered unrepresentative by RMG and not used for reaction generation. For heteroatom molecules with lone pairs, more delocalization pathways are now recognized by RMG. To address the increase in computational requirements for handling additional resonance pathways and structures, as well as to identify the representative localized structures, a heuristic-based filtration algorithm will identify representative resonance structures on-the-fly. This approach was shown to correspond well to quantum calculations.[24]

Methods for estimating parameters in the model

Parameter estimation is possibly the most important step in mechanism generation, especially for flux-based algorithms like the one used in RMG. Because the criteria for selecting species to add into the model depends on the calculated reaction flux to those species, thermochemistry and rate constant predictions must not only be accurate for important species, but they must be reasonably correct for unimportant species, so that they can be properly neglected. The estimation algorithms rely on data which have been collected and stored in the RMG-database.[28] This release of RMG includes both newly added data and new estimation algorithms.

Thermochemistry

For thermochemistry estimation, RMG relies primarily on group additivity, where the thermochemistry for a molecule is derived from the sum of contributions from each heavy atom.[29,30] However, a major limitation of standard group additivity is that only local fea-

tures are captured; longer range effects such as steric interactions and ring strain must be treated separately.

For steric interactions, RMG previously included a limited set of gauche (i.e. 1,4) and 1,5-interactions. In the current version, these corrections have been re-organized into cyclic and non-cyclic non-nearest-neighbor interactions, following the addition of a new set of group additivity values for ring substituents by Ince et al. [31,32]

Ring strain can substantially affect the thermochemistry of many cyclic and polycyclic species. A previous limitation of the group additivity algorithm was that ring strain corrections would only be applied if there was an exact match to the molecule. To address this, a new estimation algorithm was developed to provide an estimate for the ring strain of any molecule based on a heuristic algorithm which decomposes the molecule into mono- and bicyclic substructures. [33]

Furthermore, the group additivity estimator in RMG has been significantly expanded for sulfur compounds, with the addition of 200 new group values for various C/H/O/S groups. [34] These values were fitted from a collection of thermochemical data derived from quantum chemistry calculations.

Going beyond group additivity, RMG v3.0 also includes an updated neural network based thermochemistry estimator, developed using the *chemprop* package[35] for molecular property prediction. Many molecular property prediction models are based on DFT data, including the previous version of the RMG thermochemistry estimator, [36] because they are readily available in large databases or can be calculated with low computational cost. Since RMG strongly benefits from more accurate predictions, the new thermochemistry estimator was designed using a transfer learning approach that is able to learn accurate models from small high-quality data sets composed of experimental and coupled cluster calculations. [37] As described in the *chemprop* publication, [35] the deep-learning models use a message passing neural network (MPNN) to encode molecular graphs into fixed-length feature vectors which are passed through additional fully-connected neural network layers to make the

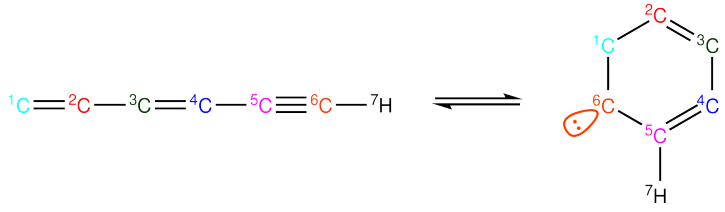
thermochemistry predictions. Instead of using the featurization for atoms and bonds implemented by *chemprop*, we removed features that depend on resonance structure and added ring membership features, which we have shown to be beneficial.[36,37] Two separate models were trained, one to predict enthalpies of formation and one to predict entropy and heat capacities simultaneously.

Kinetics

Kinetics families

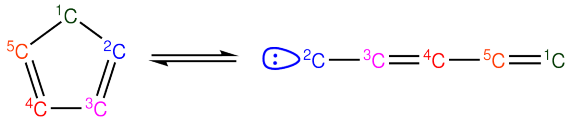
New kinetics families have been implemented in RMG to allow automatic enumeration of new reaction pathways. All of the new families which have been added since RMG v1.0.0 are shown in Table 2. A complete list of all families can be found on GitHub.[38] These new kinetics families include reactions involved in the propargyl recombination pathway to benzene formation,[12] peroxide reactions relevant in liquid phase oxidation chemistry, surface reaction types for heterogeneous catalysis simulations,[18] and a few other reactions types which have been found to be important for various systems.

Table 2: New kinetics families available in RMG v3.0

Propargyl recombination reaction families
`6_membered_central_C-C_shift` 
`Concerted_Intra_Diels_alder_monocyclic_1,2_shiftH` 

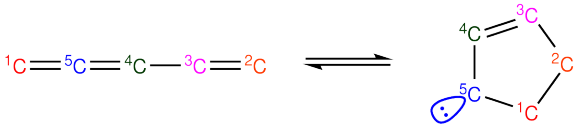
(Continued)

Cyclopentadiene_scission

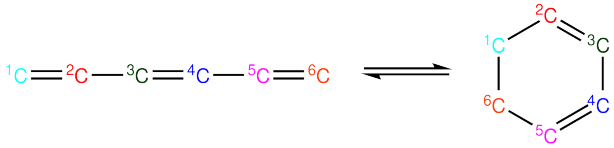


Intra_2+2_cycloaddition_Cd

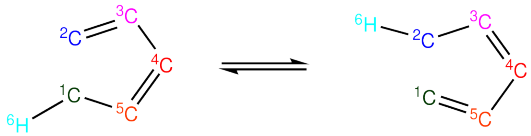
Intra_5_membered_conjugated_C=C_C=C_addition



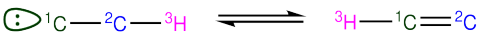
Intra_Diels_alder_monocyclic



Intra_ene_reaction (Previously H_shift_cyclopentadiene)



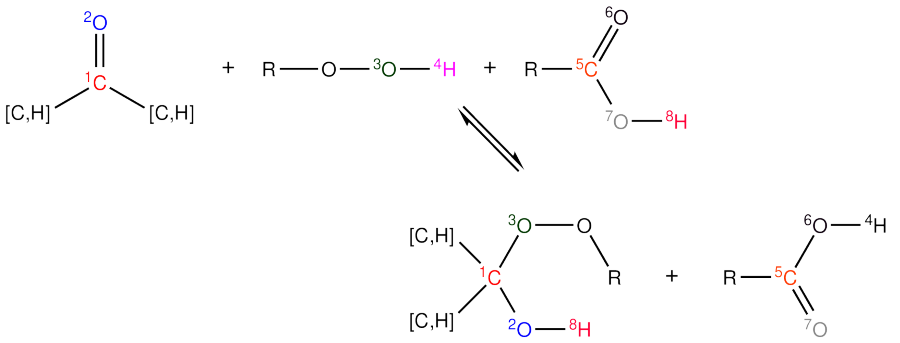
Singlet_Carbene_Intra_Disproportionation



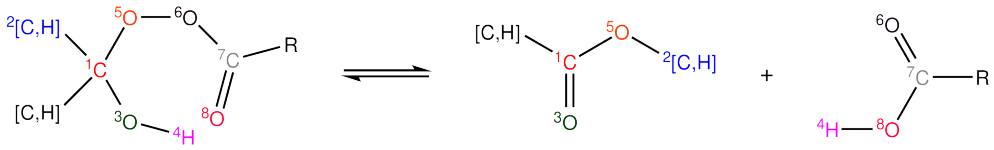
Liquid phase peroxide oxidation reaction families

(Continued)

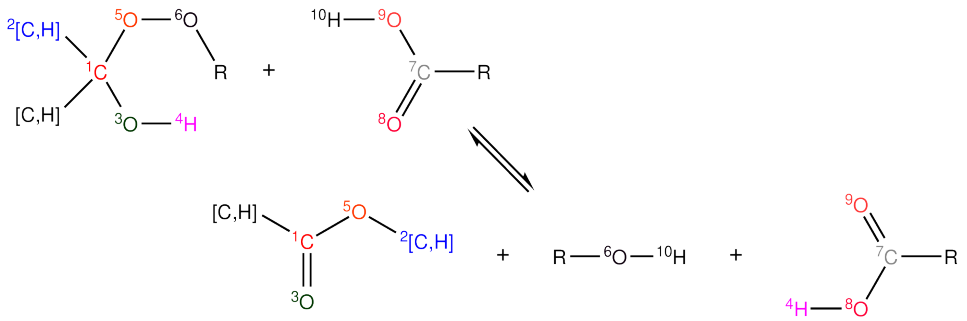
Baeyer-Villiger_step1_cat



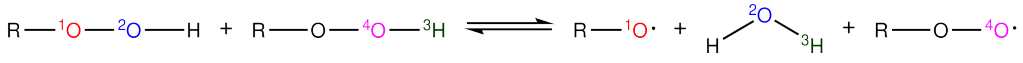
Baeyer-Villiger_step2



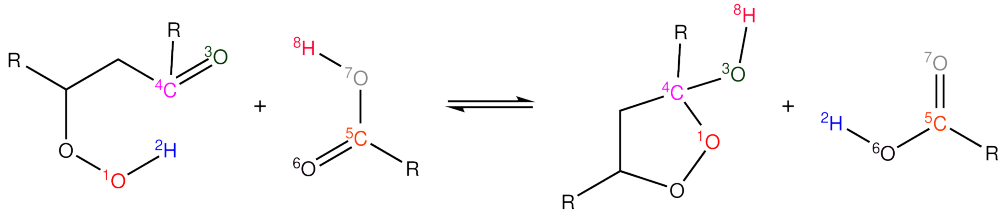
Baeyer-Villiger_step2_cat



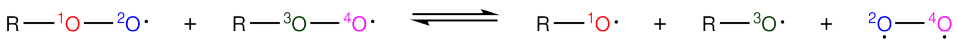
Bimolec_Hydroperoxide_Decomposition



Korcek_step1_cat

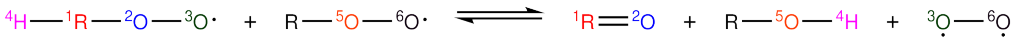


Peroxyl_Disproportionation



(Continued)

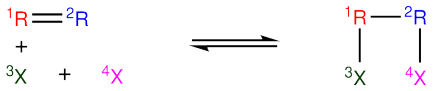
Peroxyl_Termination



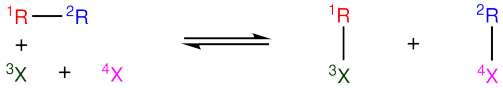
Surface reaction families

Surface_Abstraction

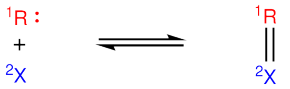
Surface_Adsorption_Bidentate



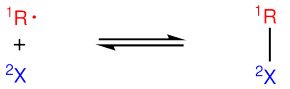
Surface_Adsorption_Dissociative



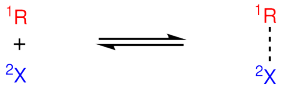
Surface_Adsorption_Double



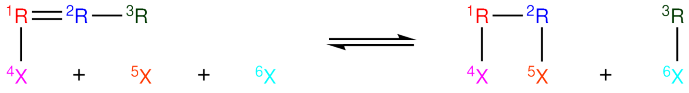
Surface_Adsorption_Single



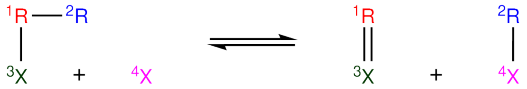
Surface_Adsorption_vdW



Surface_Bidentate_Dissociation

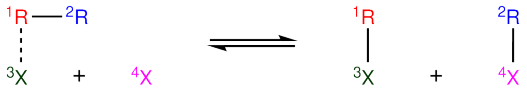


Surface_Dissociation

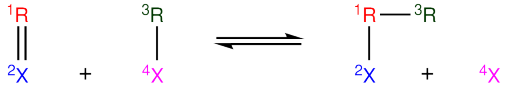


(Continued)

Surface_Dissociation_vdW

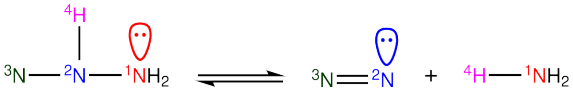


Surface_Recombination

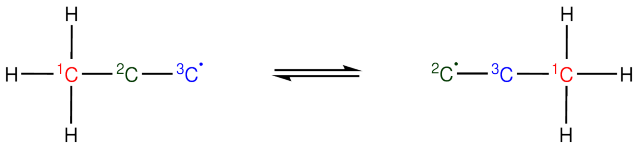


Other new reaction families

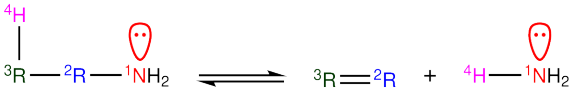
1,2_NH3_elimination



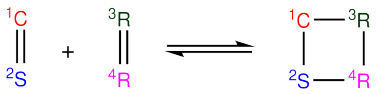
1,2_shiftC



1,3_NH3_elimination

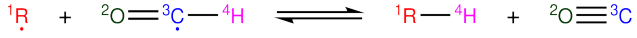


2+2_cycloaddition_CS

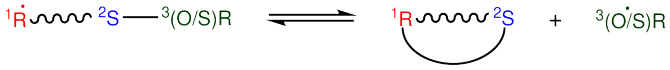


Birad_R_Recombination (Previously 0a_R_Recombination)

CO_Disproportionation

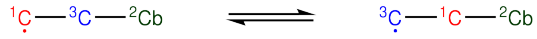


Cyclic_Thioether_Formation

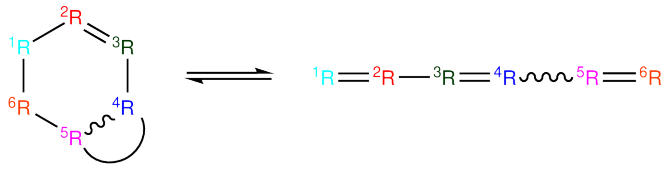


(Continued)

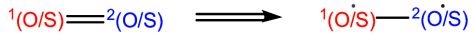
`Intra_R_Add_Exo_scission`



`Intra_Retro_Diels_alder_bicyclic` (Previously `Intra_Diels_alder`)



`Singlet_Val6_to_triplet`



Automated tree generation

One major challenge with the original kinetics family format was the need to manually maintain and design the tree structure for each family. When adding new training reactions, it is often necessary to extend the tree with new group structures in order to optimize the utilization of training reactions in generating new rate rules.

The solution which is being introduced in RMG v3.0 is the capability of automatically generating the tree. The new method uses machine learning approaches to automatically generate a decision tree based on the available training reactions. Starting with a generic reaction template, new groups are generated based on pre-defined types of extensions, e.g., adding an atom, adding a bond, specifying an element, etc. An optimal extension is chosen at each level of the tree by determining information gain based on the reduction in reaction rate variance. More details of the algorithm will be described in a separate publication.

In the v3.0 release, the `R_Recombination` family has been updated with an automatically generated tree. Updates to other reaction families can be expected in upcoming releases.

Performance improvement

Parallel computing

Many of the described additions to RMG may facilitate the construction of high-fidelity kinetic models, but they also increase computational demands. Consequently, without addressing the performance of the algorithms, many of the current and upcoming features will be available in theory, but not affordable in practice.

To address this challenge, a software package like RMG should make use of up-to-date computational hardware to improve its performance without sacrificing accuracy of the generated mechanisms. Computational hardware development, more specifically, new chip designs allow for the addition of several cores to a single processor. Furthermore, each core might allocate a number of threads that can execute parts of a software in parallel and therefore, reduce execution time.

In the case of RMG, parallelization is challenging to implement since the core algorithm described in detail by Gao et al.[7] is iterative in nature, i.e., tasks must be performed in order because they rely on the results or prior tasks. However, there are certain portions of the algorithm which are more amenable to parallelization. In RMG v3.0, parallelization has been completely revamped using the built-in `multiprocessing` module in Python, providing parallel processing support for reaction generation and quantum calculations for the QMTP (Quantum Mechanics for Thermochemical Properties) module.[39]

Molecule comparison

One task which can require substantial computing time in RMG is molecule comparison, which is done to identify if two molecules in RMG are the same chemical species. Part of the challenge is because RMG uses localized resonance structures to represent molecules, so simply comparing two structures may not be sufficient to determine whether or not they are the same. Instead, all of the resonance structures must be compared. Therefore, the

standard approach to comparing molecules was to generate all resonance structures for the two species and comparing them to each other using graph isomorphism. To confirm that two molecules are the same, the comparison can return as soon as a matching pair of resonance structures is found. However, to confirm that two molecules are different, all combinations of resonance structures must be checked.

The previous approach was very time-consuming, especially when considering resonance structure generation. A timing comparison of various methods for comparing molecules is shown in Figure 5 for five test cases of comparing identical or different molecules. The first (blue) bar shows timing for resonance structure generation followed by graph isomorphism, and it's clear that the resonance structure generation task increases the total time by over an order-of-magnitude, even for species without resonance.

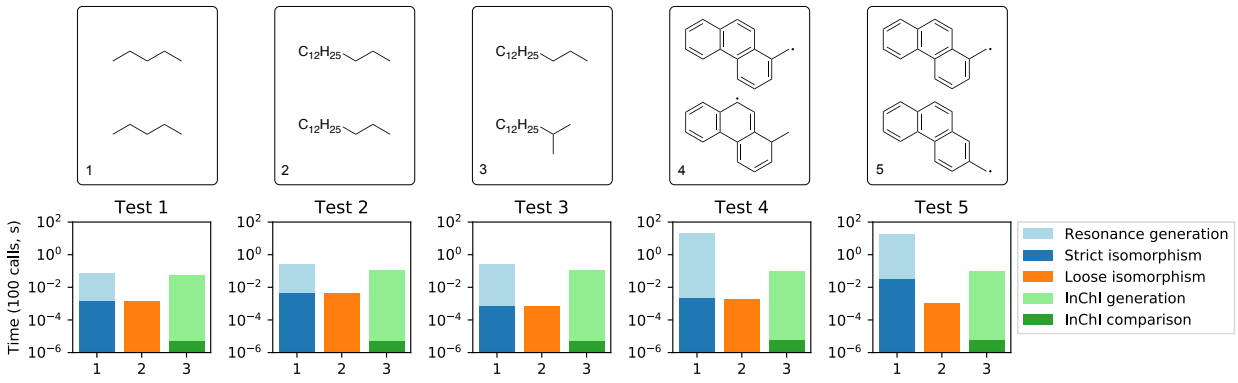


Figure 5: Comparison of walltime for 100 calls of molecule comparison methods in RMG for five different cases. Method 1 is resonance structure generation followed by strict isomorphism. Method 2 is loose isomorphism. Method 3 is InChI generation and comparison.

A newly implemented isomorphism method, referred to here as "loose isomorphism," relies on ignoring electron-related features, such as radicals, lone pairs, and bond orders. Multiplicity is considered in order to distinguish electronic states. Charge is not yet considered, since RMG does not currently support ions. The purpose is to have an isomorphism approach which is independent of resonance structures and only focuses on the atom arrangement. This eliminates the need to generate resonance structures, and both positive and negative results can be determined by comparing a single pair of structures. This new approach can

identify identical molecules more reliably than strict isomorphism because it avoids any limitations in the resonance generation algorithm. For example, prior to the implementation of an algorithm for benzyne resonance, loose isomorphism could correctly identify the two resonance structures of benzene as being the same molecule while strict isomorphism could not. The timing for this method is shown by the second (orange) bar in Figure 5. We see that this method is almost identical in performance to normal isomorphism, with the main difference being faster identification of different molecules with many resonance structures, as demonstrated by Test 5. Additionally, there is a guaranteed performance improvement because resonance structure generation is avoided.

A third option which also has significant potential is to compare the International Chemical Identifier (InChI) for various molecules. An InChI is a string identifier for a molecule which is designed to be independent of resonance structures and would therefore give the same result as the loose isomorphism method, although multiplicity would still need to be considered separately because InChI does not account for electronic states. Additionally, string comparison is extremely fast. Unfortunately, InChI generation time is non-trivial. The third (green) bar in Figure 5 shows the time required for generating InChI strings for two molecules and comparing them. Though the string comparison is extremely fast as expected, InChI generation makes the overall process take longer than graph isomorphism. For completeness, we note that SMILES comparison does not meet our needs because each resonance structure would have a different SMILES string.

It is important to note that these timings are not completely representative of actual operation. Importantly, resonance structures and InChI strings can be cached, such that they only need to be generated once. Then subsequent comparisons would require much less time. However, a large portion of comparisons in RMG are with newly generated molecules, where the data would always need to be generated. As a result, the true cost of these comparisons would be in between the total time and just the comparison time.

In RMG v3.0, most molecule comparisons have been changed to use loose isomorphism

because it is a guaranteed improvement over resonance structure generation plus strict isomorphism. However, InChI comparison should be considered in the future if InChI generation speed is improved.

Development practices

With continued growth of the RMG development team and user-base, good software development practices have become increasingly important. In recent years, additional emphasis has been placed on implementing best-practices for open-source software development. All RMG source code is publicly available on GitHub.[40] Code review and continuous integration testing are emphasized as part of the development workflow, which has been formalized via official contributor guidelines.[41] Elements of git-flow[42] and semantic versioning[43] have also been implemented into the development workflow to improve version release planning.

Conclusions

RMG v3.0 is now available, and we recommend existing users to update their installations to take advantage of new features. Linux and MacOS are supported natively, and Windows is supported via the Windows Subsystem for Linux (WSL). Compared to RMG v1.0.0, there are many new features and substantial improvements across all aspects of the software. Python 3 support ensures that RMG is up to date with the latest scientific packages and will be for the foreseeable future. New chemistry features like surface mechanism generation and isotopic mechanism generation enable application of RMG to more systems than ever before. Uncertainty analysis provides new ways to analyze models to quantify the overall uncertainty in a model and identify the parameters which contribute most to that uncertainty. Fundamental improvements to molecular representation in the form of new atom types and resonance transformations work together to improve the the accuracy of the localized molecular representations. Parameter estimation, as the key to generating good models, has been

improved via expansion of the database as well as addition of new algorithms like the neural network thermochemistry estimator. Finally, performance improvement is always an ongoing focus, and the recent implementation of parallel computing and improved molecule isomorphism comparison are steps towards faster model generation.

All of the developments mentioned here, and countless others which can be explored in the detailed RMG release notes,[44] have greatly improved the accuracy, robustness, and applicability of RMG to modeling various chemical systems. RMG development is more active than it has been at any point in the past, which promises to continue bringing new and exciting improvements. For example, ongoing development of automated high-throughput quantum calculations for both thermochemistry and kinetics, leading-edge machine learning methods for parameter prediction, and new model expansion algorithms to complement species selection by flux are leading toward construction of even more accurate models using automatic mechanism generation.

Acknowledgement

The authors thank past and current RMG developers, specifically Connie Gao and Nick Vandewiele for helpful discussions.

Alon Grinberg Dana acknowledges financial support from The Nancy & Stephen Grand Technion Energy Program (GTEP) and from The Mortimer B. Zuckerman STEM Leadership Program. Mark Jacob Goldman acknowledges financial support from the National Science Foundation Graduate Research Fellowship grant number 1122374. Agnes Jocher acknowledges financial support from the DFG Research Fellowship JO 1526/1–1.

The primary financial support for the work reported in this paper came from the Gas Phase Chemical Physics Program of the U.S. Department of Energy, Office of Basic Energy Sciences, Division of Chemical Sciences, Geosciences, and Biosciences (under Award number DE-SC0014901). The work was partially supported by the U.S. Department of Energy,

Office of Science, Basic Energy Sciences, under Award number 0000232253, as part of the Computational Chemical Sciences Program.

Additional support by subcontract 7F-30180 to MIT from UC Chicago Argonne LLC is also gratefully acknowledged. It is a component of the Exascale Computing Project (ECP), Project Number 17-SC-20-SC, a collaborative effort of two DOE organizations, the Office of Science and the National Nuclear Security Administration, responsible for the planning and preparation of a capable exascale ecosystem including software, applications, hardware, advanced system engineering, and early test bed platforms to support the nation's exascale computing imperative.

References

(1) Vandewiele, N. M.; Van Geem, K. M.; Reyniers, M.-F.; Marin, G. B. Genesys: Kinetic model construction using chemo-informatics. *Chemical Engineering Journal* **2012**, *207-208*, 526 – 538, 22nd International Symposium on Chemical Reaction Engineering (ISCRE 22).

(2) Broadbelt, L.; Stark, S.; Klein, M. Computer generated reaction modelling: Decomposition and encoding algorithms for determining species uniqueness. *Computers & Chemical Engineering* **1996**, *20*, 113 – 129.

(3) Van de Vijver, R.; Vandewiele, N. M.; Bhoorasingh, P. L.; Slakman, B. L.; Seyedzadeh Khanshan, F.; Carstensen, H.-H.; Reyniers, M.-F.; Marin, G. B.; West, R. H.; Van Geem, K. M. Automatic Mechanism and Kinetic Model Generation for Gas- and Solution-Phase Processes: A Perspective on Best Practices, Recent Advances, and Future Challenges. *International Journal of Chemical Kinetics* **2015**, *47*, 199–231.

(4) Rangarajan, S.; Bhan, A.; Daoutidis, P. Language-oriented rule-based reaction network

generation and analysis: Description of RING. *Computers & Chemical Engineering* **2012**, *45*, 114 – 123.

(5) Warth, V.; Battin-Leclerc, F.; Fournet, R.; Glaude, P.; Côme, G.; Scacchi, G. Computer based generation of reaction mechanisms for gas-phase oxidation. *Computers & Chemistry* **2000**, *24*, 541 – 560.

(6) Blurock, E. S. Reaction: System for Modeling Chemical Reactions. *Journal of Chemical Information and Computer Sciences* **1995**, *35*, 607–616.

(7) Gao, C. W.; Allen, J. W.; Green, W. H.; West, R. H. Reaction Mechanism Generator: Automatic construction of chemical kinetic mechanisms. *Comput. Phys. Commun.* **2016**, *203*, 212–225.

(8) Grinberg Dana, A.; Buesser, B. A.; Merchant, S. S.; Green, W. H. Automated Reaction Mechanism Generation Including Nitrogen as a Heteroatom. *Int. J. Chem. Kinet.* **2018**, *50*, 243–258.

(9) Class, C. A.; Liu, M.; Vandeputte, A. G.; Green, W. H. Automatic mechanism generation for pyrolysis of di-tert-butyl sulfide. *Phys. Chem. Chem. Phys.* **2016**, *18*, 21651–21658.

(10) Lai, L.; Gudiyella, S.; Liu, M.; Green, W. H. Chemistry of Alkylaromatics Reconsidered. *Energy & Fuels* **2018**, *32*, 5489–5500.

(11) Zhang, P.; Yee, N. W.; Filip, S. V.; Hetrick, C. E.; Yang, B.; Green, W. H. Modeling study of the anti-knock tendency of substituted phenols as additives: an application of the reaction mechanism generator (RMG). *Phys. Chem. Chem. Phys.* **2018**, *20*, 10637–10649.

(12) Chu, T.-C.; Buras, Z. J.; Oßwald, P.; Liu, M.; Goldman, M. J.; Green, W. H. Modeling

of aromatics formation in fuel-rich methane oxy-combustion with an automatically generated pressure-dependent mechanism. *Phys. Chem. Chem. Phys.* **2019**, *21*, 813–832.

(13) Blondal, K.; Jelic, J.; Mazeau, E.; Studt, F.; West, R. H.; Goldsmith, C. F. Computer-Generated Kinetics for Coupled Heterogeneous/Homogeneous Systems: A Case Study in Catalytic Combustion of Methane on Platinum. *Industrial & Engineering Chemistry Research* **2019**, *58*, 17682–17691.

(14) Green, W. H. In *Mathematical Modelling of Gas-Phase Complex Reaction Systems: Pyrolysis and Combustion*; Faravelli, T., Manenti, F., Ranzi, E., Eds.; Computer Aided Chemical Engineering; Elsevier, 2019; Vol. 45; pp 259 – 294.

(15) Green, W. H. Moving from postdictive to predictive kinetics in reaction engineering. *AIChE Journal* **2020**, *66*, e17059.

(16) Allen, J. W.; Gao, C. W. PyDAS: A Python wrapper for the DASSL, DASPK, and DASKR differential algebraic system solvers. 2016; `https://github.com/ReactionMechanismGenerator/PyDAS`.

(17) Allen, J. W. PyDQED: A Python wrapper for the DQED constrained nonlinear optimization code. 2011; `https://github.com/ReactionMechanismGenerator/PyDQED`.

(18) Goldsmith, C. F.; West, R. H. Automatic Generation of Microkinetic Mechanisms for Heterogeneous Catalysis. *J. Phys. Chem. C* **2017**, *121*, 9970–9981.

(19) Gao, C. W.; Liu, M.; Green, W. H. Uncertainty analysis of correlated parameters in automated reaction mechanism generation. *International Journal of Chemical Kinetics* **2020**, *52*, 266–282.

(20) Conrad, P. R.; Parno, M. D.; Davis, A. D.; Marzouk, Y. M. MIT Uncertainty Quantification Library (MUQ 2). 2019; `http://muq.mit.edu`.

(21) Goodwin, D. G.; Speth, R. L.; Moffat, H. K.; Weber, B. W. Cantera: An Object-oriented Software Toolkit for Chemical Kinetics, Thermodynamics, and Transport Processes. \url{https://www.cantera.org}, 2018.

(22) Goldman, M. J.; Vandewiele, N. M.; Ono, S.; Green, W. H. Computer-generated isotope model achieves experimental accuracy of filiation for position-specific isotope analysis. *Chem. Geol.* **2019**, *514*, 1–9.

(23) Julien, M.; Goldman, M. J.; Liu, C.; Horita, J.; Boreham, C. J.; Yamada, K.; Green, W. H.; Yoshida, N.; Gilbert, A. Intramolecular 13C isotope distributions of butane from natural gases. *Chemical Geology* **2020**, *541*, 119571.

(24) Grinberg Dana, A.; Liu, M.; Green, W. H. Automated chemical resonance generation and structure filtration for kinetic modeling. *Int. J. Chem. Kinet.* **2019**, *51*, 760–776.

(25) Liu, M.; Green, W. H. Capturing Aromaticity in Automatic Mechanism Generation Software. *Proc. Combust. Inst.* **2019**, *37*, 575–581.

(26) Clar, E.; Zander, M. 1:12-2:3-10:11-Tribenzoperylene. *J. Chem. Soc.* **1958**, 1861–1864.

(27) Solà, M. Forty years of Clar's aromatic π-sextet rule. *Front. Chem.* **2013**, *1*, 22.

(28) RMG-database GitHub. 2019; `https://github.com/ReactionMechanismGenerator/RMG-database`.

(29) Benson, S. W.; Buss, J. H. Additivity Rules for the Estimation of Molecular Properties. Thermodynamic Properties. *J. Chem. Phys.* **1958**, *29*, 546–572.

(30) Benson, S. W. *Thermochemical kinetics : methods for the estimation of thermochemical data and rate parameters.*; New York : Wiley, c1976., 1976.

(31) Ince, A.; Carstensen, H.-H.; Reyniers, M.-F.; Marin, G. B. First-principles based group additivity values for thermochemical properties of substituted aromatic compounds. *AIChE Journal* **2015**, *61*, 3858–3870.

(32) Ince, A.; Carstensen, H.-H.; Sabbe, M.; Reyniers, M.-F.; Marin, G. B. Group additive modeling of substituent effects in monocyclic aromatic hydrocarbon radicals. *AIChE Journal* **2017**, *63*, 2089–2106.

(33) Han, K.; Jamal, A.; Grambow, C. A.; Buras, Z. J.; Green, W. H. An Extended Group Additivity Method for Polycyclic Thermochemistry Estimation. *Int. J. Chem. Kinet.* **2018**, *50*, 294–303.

(34) Gillis, R. J.; Green, W. H. Thermochemistry Prediction and Automatic Reaction Mechanism Generation for Oxygenated Sulfur Systems: A Case Study of Dimethyl Sulfide Oxidation. *ChemSystemsChem* **2020**, syst.201900051.

(35) Yang, K.; Swanson, K.; Jin, W.; Coley, C.; Eiden, P.; Gao, H.; Guzman-Perez, A.; Hopper, T.; Kelley, B.; Mathea, M.; Palmer, A.; Settels, V.; Jaakkola, T.; Jensen, K.; Barzilay, R. Analyzing Learned Molecular Representations for Property Prediction. *J. Chem. Inf. Model.* **2019**, *59*, 3370–3388.

(36) Li, Y.-P.; Han, K.; Grambow, C. A.; Green, W. H. Self-Evolving Machine: A Continuously Improving Model for Molecular Thermochemistry. *J. Phys. Chem. A* **2019**, *123*, 2142–2152.

(37) Grambow, C. A.; Li, Y.-P.; Green, W. H. Accurate Thermochemistry with Small Data Sets: A Bond Additivity Correction and Transfer Learning Approach. *J. Phys. Chem. A* **2019**, *123*, 5826–5835.

(38) List of RMG Reaction Families. 2020; `https://github.com/ReactionMechanismGenerator/RMG-database/blob/master/families/rmg_reaction_families.pdf`.

(39) Jocher, A.; Vandewiele, N. M.; Han, K.; Liu, M.; Gao, C. W.; Gillis, R. J.; Green, W. H. Scalability strategies for automated reaction mechanism generation. *Comput. Chem. Eng.* **2019**, 106578.

(40) RMG-Py GitHub. 2019; https://github.com/ReactionMechanismGenerator/RMG-Py.

(41) Liu, M.; Han, K.; Goldman, M. J.; Payne, M. A. RMG Contributor Guidelines. 2019; https://github.com/ReactionMechanismGenerator/RMG-Py/wiki/RMG-Contributor-Guidelines.

(42) Driessen, V. A successful Git branching model. 2010; https://nvie.com/posts/a-successful-git-branching-model/.

(43) Preston-Werner, T. Semantic Versioning. 2013; https://semver.org/.

(44) RMG Documentation: Release Notes. 2019; http://reactionmechanismgenerator.github.io/RMG-Py/users/rmg/releaseNotes.html.

Graphical TOC Entry