Uncertainty Analysis in RMG

Connie Gao

12/15/2016

RMG Study Group

Uncertainty propagation in kinetic systems

In a nonlinear chemical system, uncertainty of certain input parameters become magnified while others are suppressed

Input parameters $(\boldsymbol{\lambda})$

Reaction rate coefficients $(\boldsymbol{k})$
Species thermodynamics $(\boldsymbol{G})$

Output $(\boldsymbol{c})$

Product concentrations

Refining most influential uncertain parameters is fastest way to improve a model

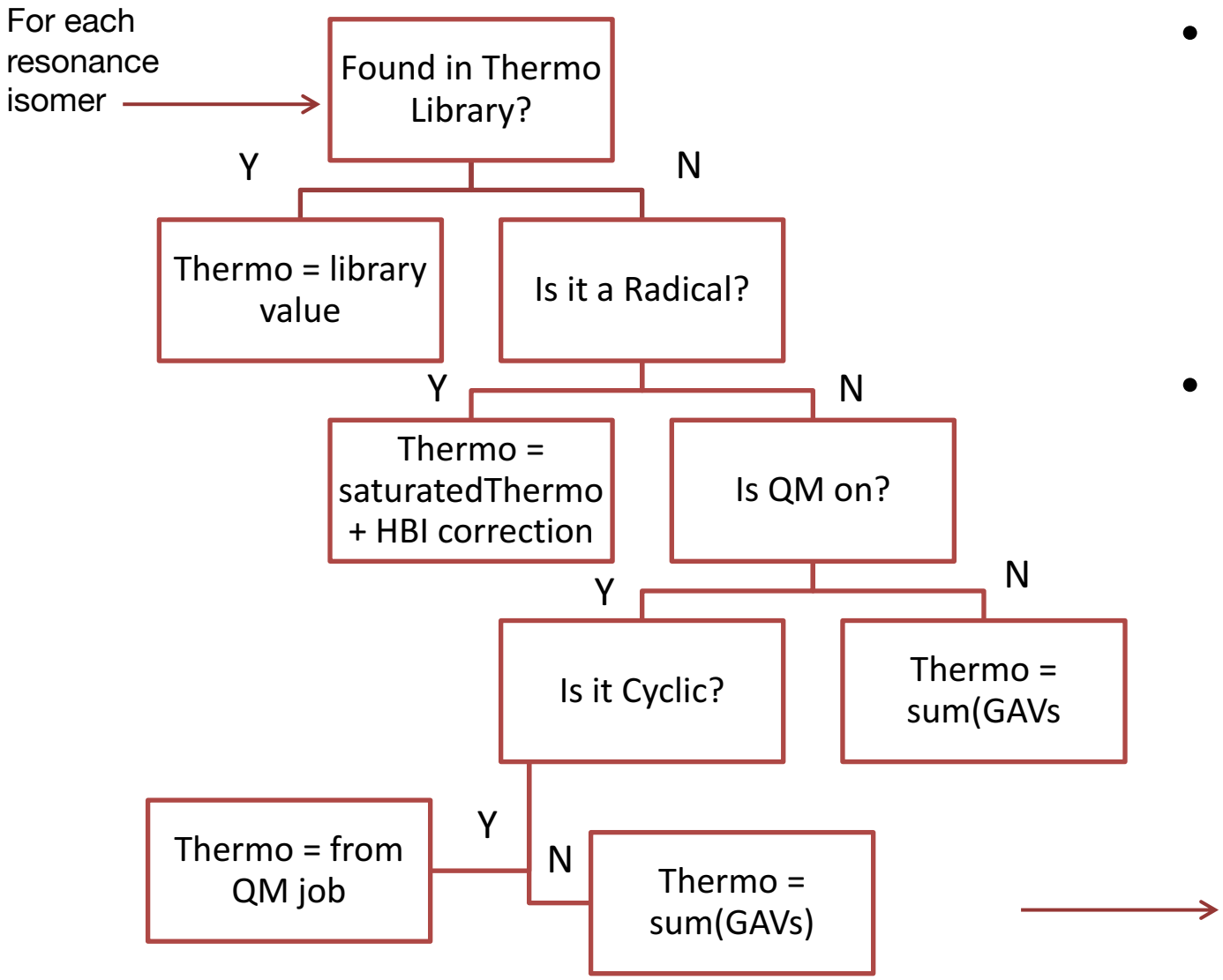
Outline

- Parameter estimation leads to uncertainty
- Local and global uncertainty analysis implementation in RMG
- Demonstration of results
- Proposal for a new model development workflow

Parameter estimation leads to uncertainty

- RMG uses many methods to estimate thermo and kinetic parameters
- Uncertainty assignment must correspond to the confidence we have in various parameter sources
  - Library kinetics derived from quantum chemistry or experiment has a very different uncertainty than a rate derived from averaging rate rules
- A parameter's uncertainty cannot be decoupled from the estimation methods used to derived that parameter!

Estimating thermochemistry: a decision tree



- 3 types of sources
  - Thermo Library
  - QM (on-the-fly quantum mechanics)
  - Group additivity (GAVs)

- But actually 2 additional types of mixed sources for radicals!
  - Thermo Library saturated value + HBI correction from GAVs
  - QM saturated value + HBI correction from GAVs

Prioritize resonance isomer thermo by rank

Thermo uncertainty assignment

- Assume a uniform uncertainty distribution in free energy

$$\mathrm{G} \in [G_{min}, G_{max}] \qquad dG = (G_{max} - G_{min})/2$$

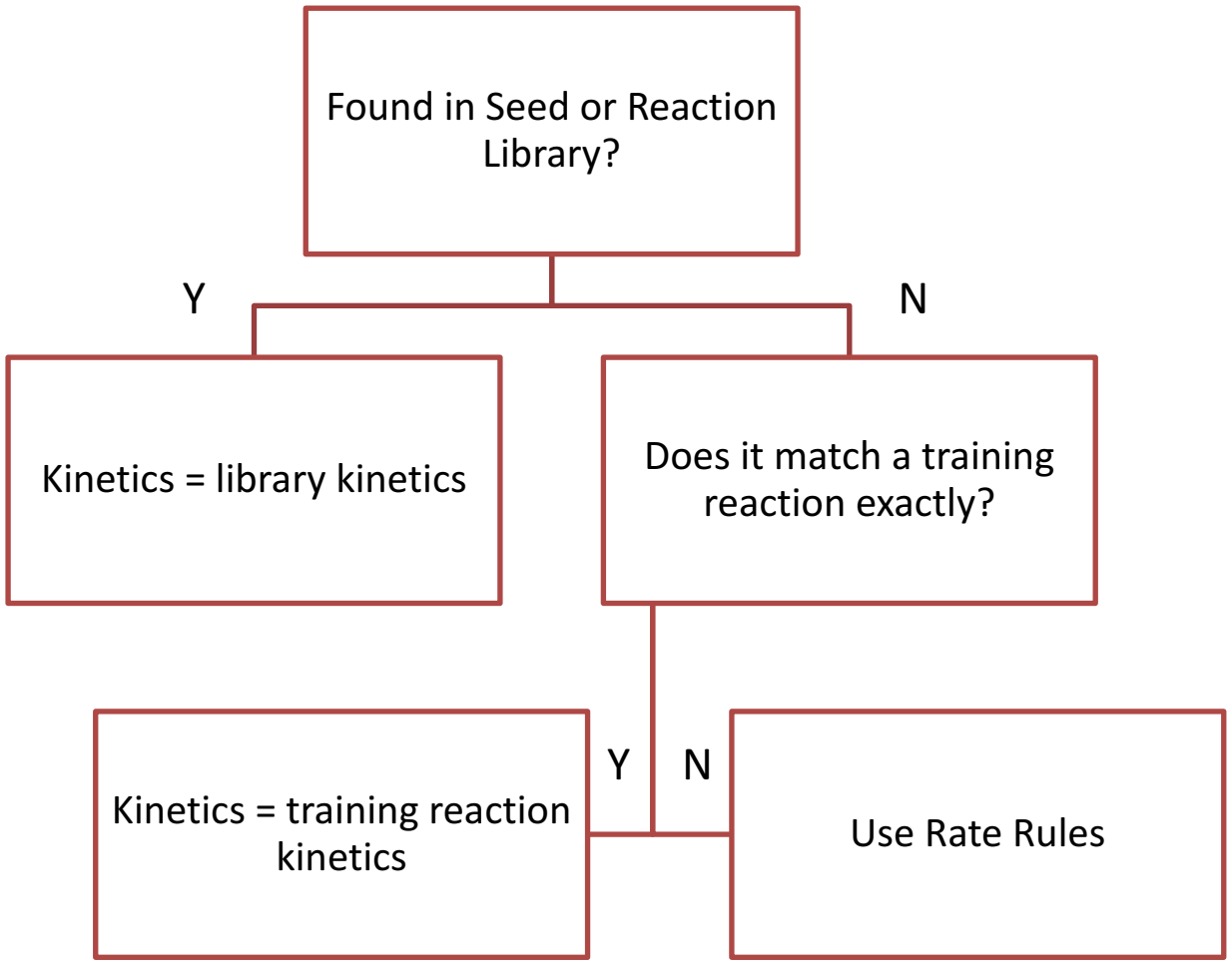
- Assign uncertainties according to what parameter sources constitute the thermo estimate

Fixed sources with true values; these errors are correlated when used estimate multiple parameters

$$(\mathrm{dG})^2 = \delta_{library}(dG_{library})^2 + \delta_{QM}(dG_{QM})^2 + \delta_{GAV}(dG_{GAV})^2 + \sum_{group}(w_{group} dG_{group})^2$$

Uncorrelated error associated with using the group additivity method for this particular thermo parameter

Estimating reaction kinetics: a decision tree

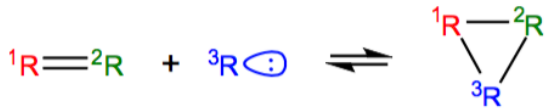


- 3 types of kinetics sources:
  - Library reaction kinetics
  - Training reaction kinetics
  - Rate rule kinetics
- But there is 1 more type of mixed source!
  - Rate rules + Rate rules originating from training reactions

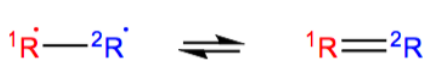
Rate Rule Kinetics

Over 40 Reaction Families in RMG

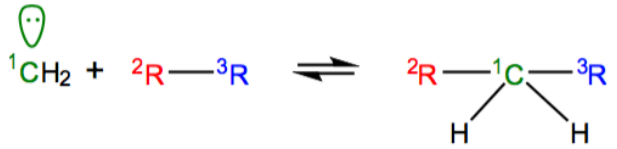
1+2_Cycloaddition



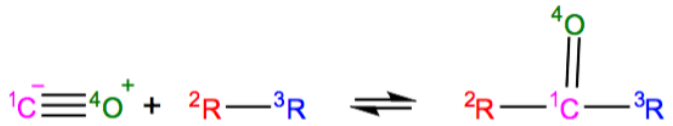
1,2-Birad_to_alkene



1,2_Insertion_carbene

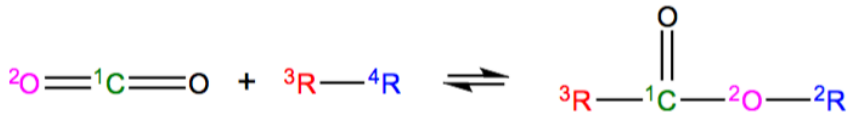


1,2_Insertion_CO



1,2_shiftS

1,3_Insertion_CO2



1,3_Insertion_ROR

1,3_Insertion_RSR

Kinetics uncertainty assignment

- Each reaction rate is assigned a loguniform uncertainty distribution

$$d\ln(k) \in [\ln(k_{min}), \ln(k_{max})] \qquad d\ln(k) = [\ln(k_{max}) - \ln(k_{min})]/2$$

- Assume that library, training, and pdep reactions have fixed uncertainties $d\ln(k_{library})$, $d\ln(k_{pdep})$, $d\ln(k_{training})$
- Rate rule estimated kinetics' uncertainty:

Error associated with using a non-exact match. Used for weighting against rates using lots of averages (N=number of rules averaged). (Distance may be a better substitute eventually)

$$[d\ln(k)]^2 = [d\ln(k_{family})]^2 + [\log_{10}(N+1) * dln(k_{non-exact})]^2 + \sum_{rule} [w_{rule} d\ln(k_{rule})]^2$$

Each family has an associated error. Some families are more sparsely populated than others and will have more error. But currently all set to the same value

Intrinsic rate rule error

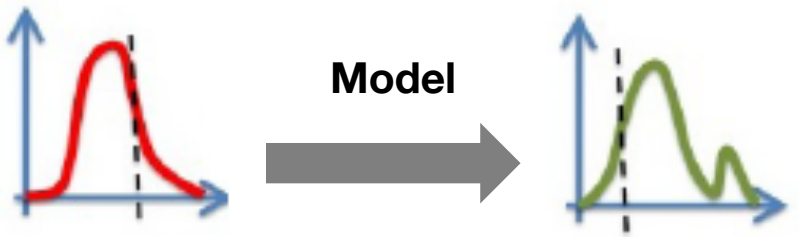
Demontration: Track parameters and assign uncertainties

- Test it live in `findParameterSourcesAndAssignUncertainies.ipynb`
- A variety of new database functions programmed to trace all parameter estimation sources and their weights
- New class `Uncertainty` contains function `extractSourcesFromModel()`

Local uncertainty propagation

$$\frac{dc}{dt} = f(c, t; \lambda)$$

$$c(t_0) = c_0$$



Input parameters $(\boldsymbol{\lambda})$

Reaction rate coefficients $(\boldsymbol{k})$
Species thermodynamics $(\boldsymbol{G})$

Output $(\boldsymbol{c})$

Concentration

$$c(t, \lambda_0 + \Delta\lambda) = c(t, \lambda_0) + \sum_j \frac{\partial c}{\partial \lambda_j}\Delta\lambda_j + \frac{1}{2}\sum_j \sum_k \frac{\partial c}{\partial \lambda_j}\frac{\partial c}{\partial \lambda_k}\Delta\lambda_j\Delta\lambda_k + \ldots$$

$$\sigma^2(c) \approx \sum_j \left(\frac{\partial c}{\partial \lambda_j}\right)^2 \sigma^2(\lambda_j)$$

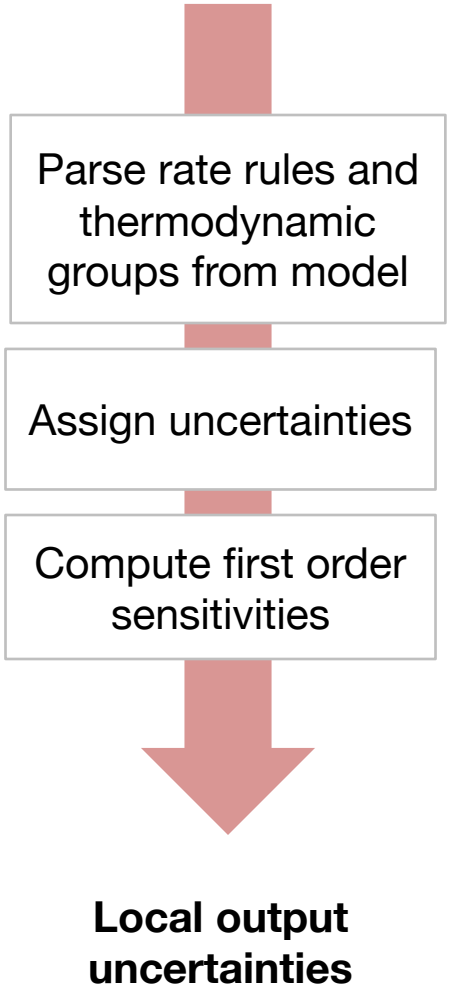
First-order sensitivity index

$$S_j = \frac{\text{Variance contributed by } \lambda_j}{\text{Total output variance}} \approx \frac{\left(\frac{\partial c}{\partial \lambda_j}\right)^2 \sigma^2(\lambda_j)}{\sigma^2(c)}$$

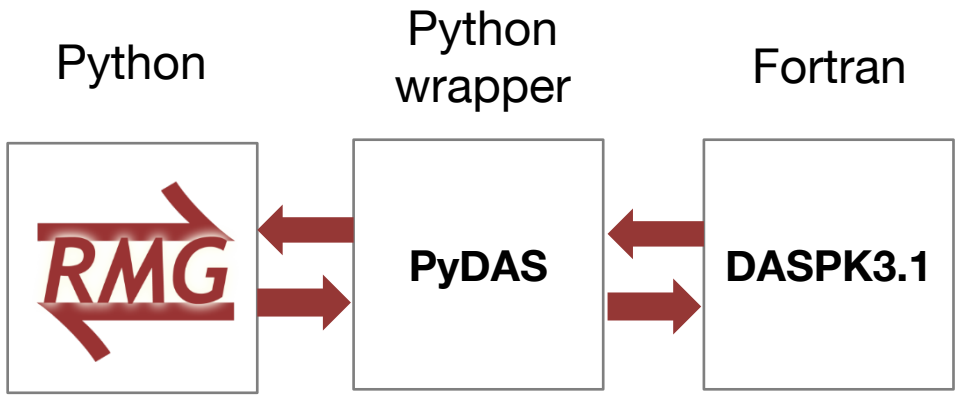
Assumptions

- Linear dependence on λ (first-order, evaluated at nominal input values)
- Independent inputs λ with no covariance

Local uncertainty propagation: implementation



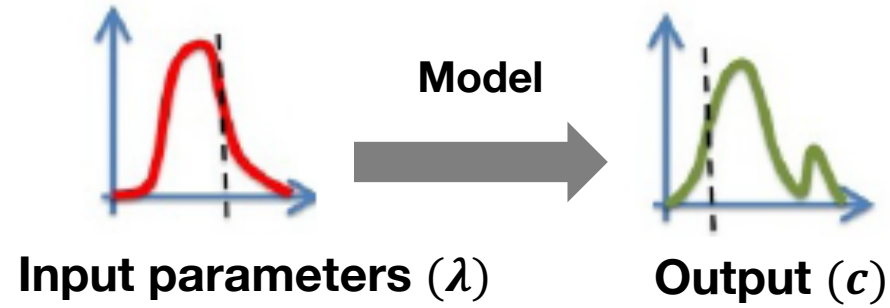
`rmgpy.tools.uncertainty`



L. Petzold, DASPK 3.1, http://www.engineering.ucsb.edu/~cse/software.html

Global uncertainty analysis

Sample from entire parameter uncertainty probability distribution



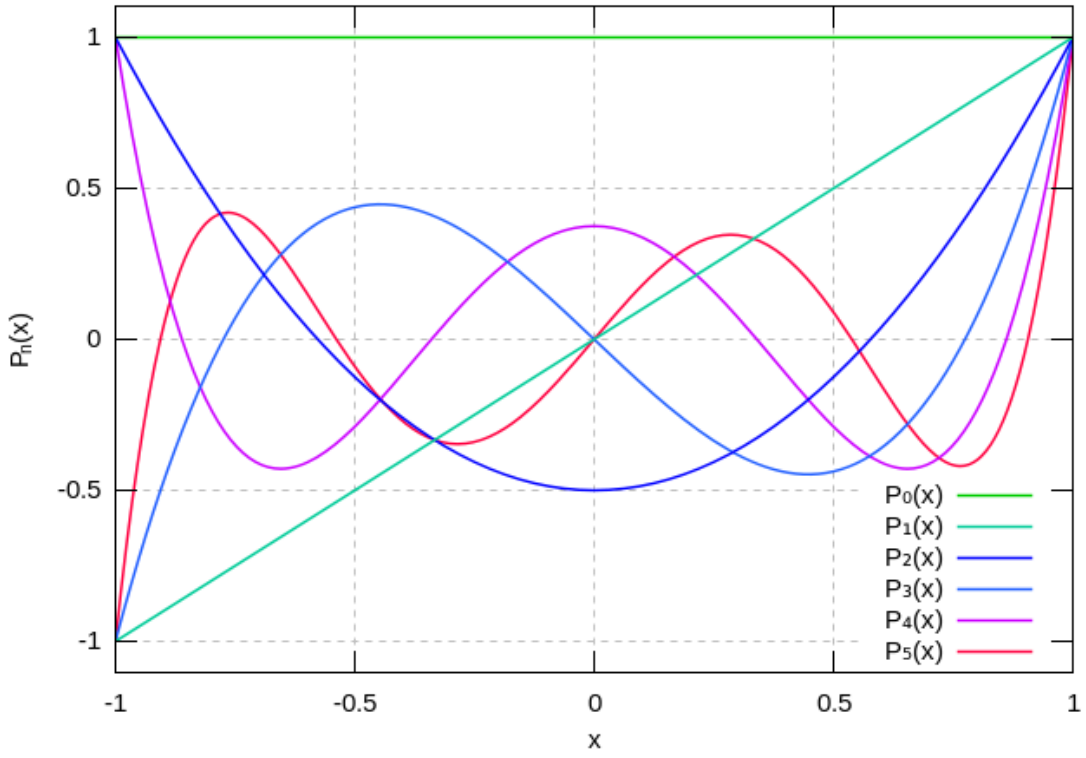
After many simulations, you can approximate the output uncertainty distribution

Methods

- Simplest but slowest: Monte Carlo
- Optimize the sampling
  - Latin hypercube sampling
  - Sobol sequences
- Basis set expansions
  - Fourier Amplitude Sensitivity Test (FAST)
  - High-dimensional model representations (HDMR)
  - **Polynomial Chaos Expansions (PCE)**

Eliminates linearity assumption, but is computationally expensive…

Legendre Polynomials



$$(n+1)P_{n+1}(\xi) = (2n+1)\xi P_n(\xi) - nP_{n-1}(\xi)$$ Three-term recurrence

$$\int_{-1}^{1} P_j(x)P_i(x)dx = \frac{2}{2n+1}\delta_{ij}$$ Orthogonality

Polynomial Chaos Expansions (PCE) *

$$c(\xi) \approx \sum_{k=0}^{P} \alpha_k \Psi_k(\xi_1, \xi_2, \ldots, \xi_n)$$

ξ is set of random, uniformly distributed independent variables $\in [-1, 1]$

Ψ_k are Legendre polynomials that form an orthogonal basis set

Compute coefficients using Galerkin projection

$$\alpha_k = \frac{\langle c\Psi_k \rangle}{\langle {\Psi_k}^2 \rangle}$$

Moments and variance of $c(\xi)$ can then be computed

Adaptive Smolyak Pseudospectral Approximations **

- Sampling performed adaptively done using a sparse grid, leading to faster convergence
- 3 ways to construct PCE:
  - adapt to fixed order
  - adapt to a heuristic error tolerance
  - select wall clock time

* D. Xiu, *SIAM J. Sci. Comput., 2002.*
** P. Conrad and Y. Marzouk, *SIAM J. Sci. Comput.*, 2013.

Global uncertainty analysis: `rmgpy.tools.muq`

MIT Uncertainty Quantification Library

http://muq.mit.edu

Package for anaconda now available for Ubuntu

Load kinetic model and initialize `ReactorModPiece`

Create `SmolyakPCEFactory`

Initialize with Legendre Polynomials Ψ_k

Sample parameter space

Random Uniform Inputs ξ

`ReactorModPiece`

Map ξ to parameters: kinetics ln(k_m) and thermodynamics G_n

Modify Cantera objects using RMG

Simulate reactor with Cantera

Cantera

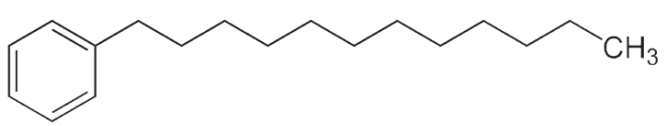
http://cantera.org

RMG

Product Concentration *c*

Obtain PCE and variance statistics

Uncertainty analysis for a toy phenyldodecane model



Pyrolysis reaction conditions

T = 350 °C

P = 35 MPa

72 hours

Model

81 species

18 group additivity values
17 thermo library values
35 independent thermodynamic parameters

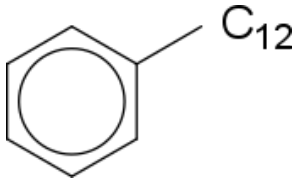
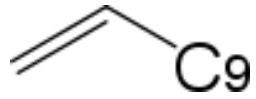
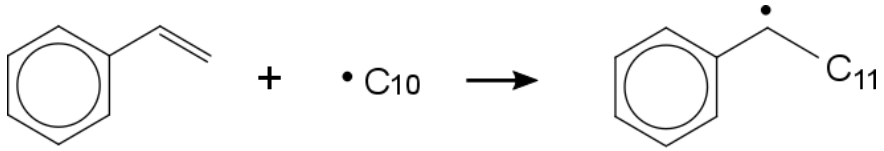
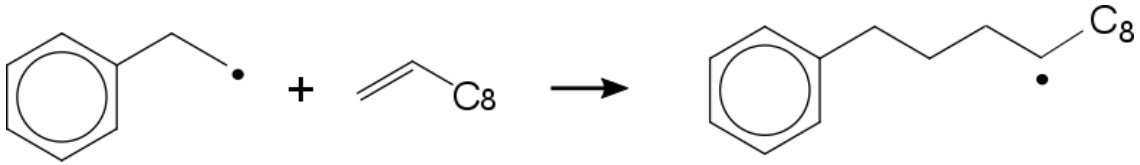
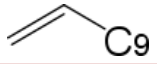
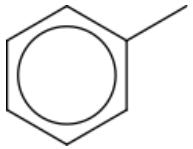
1427 reactions

4 reaction families:

- `H_Abstraction` (14 rate rules)
- `R_Recombination` (6 rate rules)
- `R_Addition_MultipleBond` (7 rate rules)
- `Disproportionation` (13 rate rules)

40 independent rate rules

Independent parameter uncertainty: global vs. local

	Phenyldodecane		Undecene	
				
	Global	**Local**	**Global**	**Local**
Mole fraction	0.187	0.146	0.141	0.108
Total variance $\sigma^2(\ln c)$	0.58	1.64	0.53	1.28
Reaction kinetics	**Sensitivity Index** S_j **(%)**			
	25.1	43.3	3.5	12.7
	18.6	22.4	2.3	7.0
Species thermochemistry	**Sensitivity Index** S_j **(%)**			
	15.1	2.3	82.4	61.0
	2.7	0.7	9.1	10.2
• C10	16.1	31.1	0.9	9.1

Correlated uncertainty propagation

- Conventional methods assume kinetic and thermo parameter uncertainties are independent, even though they are composed of **correlated** sources
- There are contributions from correlated and uncorrelated uncertainties:

$$(d\ln c_{corr,i})^2 = \sum_v \left(\frac{d\ln c_i}{d\ln k_{corr,v}} d\ln k_{corr,v}\right)^2 + \sum_w \left(\frac{d\ln c_i}{d\ln k_{res,w}} d\ln k_{res,w}\right)^2 + \sum_y \left(\frac{d\ln c_i}{dG_{corr,y}} dG_{corr,y}\right)^2 + \sum_z \left(\frac{d\ln c_i}{dG_{res,z}} dG_{res,z}\right)^2$$

$$\frac{d\ln c_i}{d\ln k_{corr,v}} = \sum_j \frac{d\ln c_i}{d\ln k_j}\frac{d\ln k_j}{d\ln k_{corr,v}} \qquad \frac{d\ln c_i}{dG_{corr,y}} = \sum_k \frac{d\ln c_i}{dG_k}\frac{dG_k}{dG_{corr,y}}$$

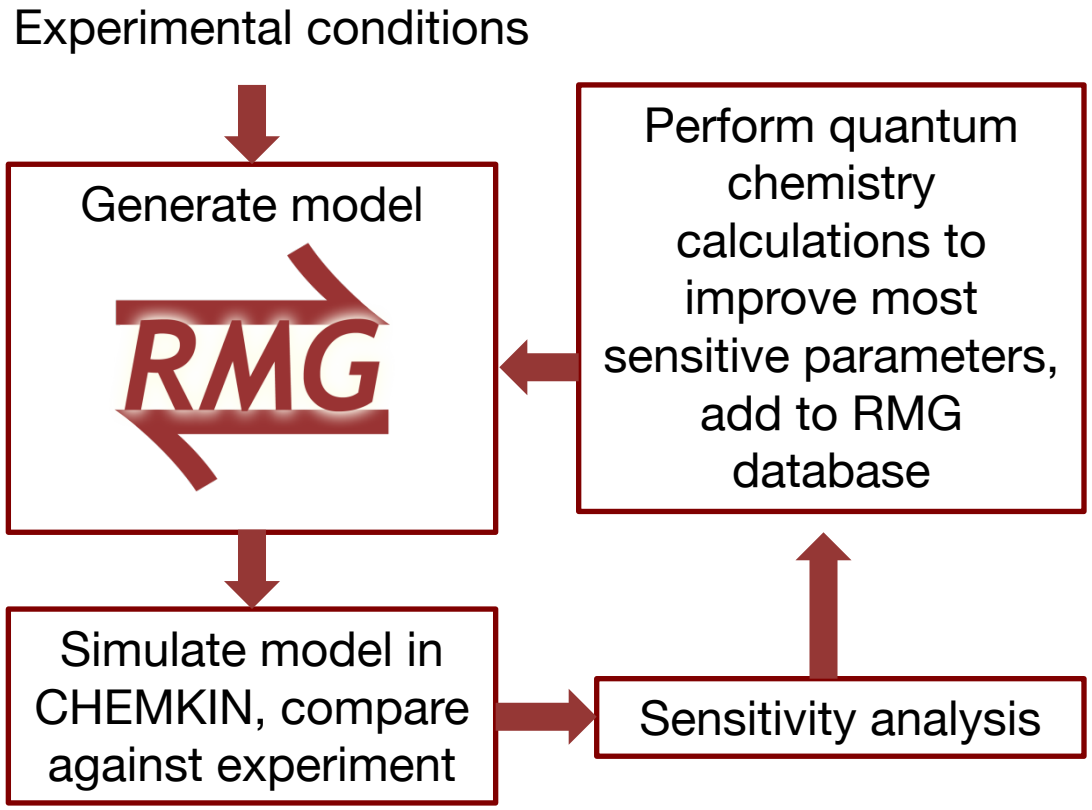
Correlated uncertainty propagation: implementation

- Classes `KineticParameterUncertainty` and `ThermoParameterUncertainty` now have the function `getPartialUncertainty()`, which can retrieve the relative contribution of uncertainty towards a parameter from a correlated source such as a rate rule
- Class `Uncertainty` has the function `assignParameterUncertainties(correlated=True)` which can now be used to assign correlated uncertainties
- Use the resulting objects that store correlated source information and partial uncertainty to propagate within the existing local and global uncertainty classes

Demonstration of results

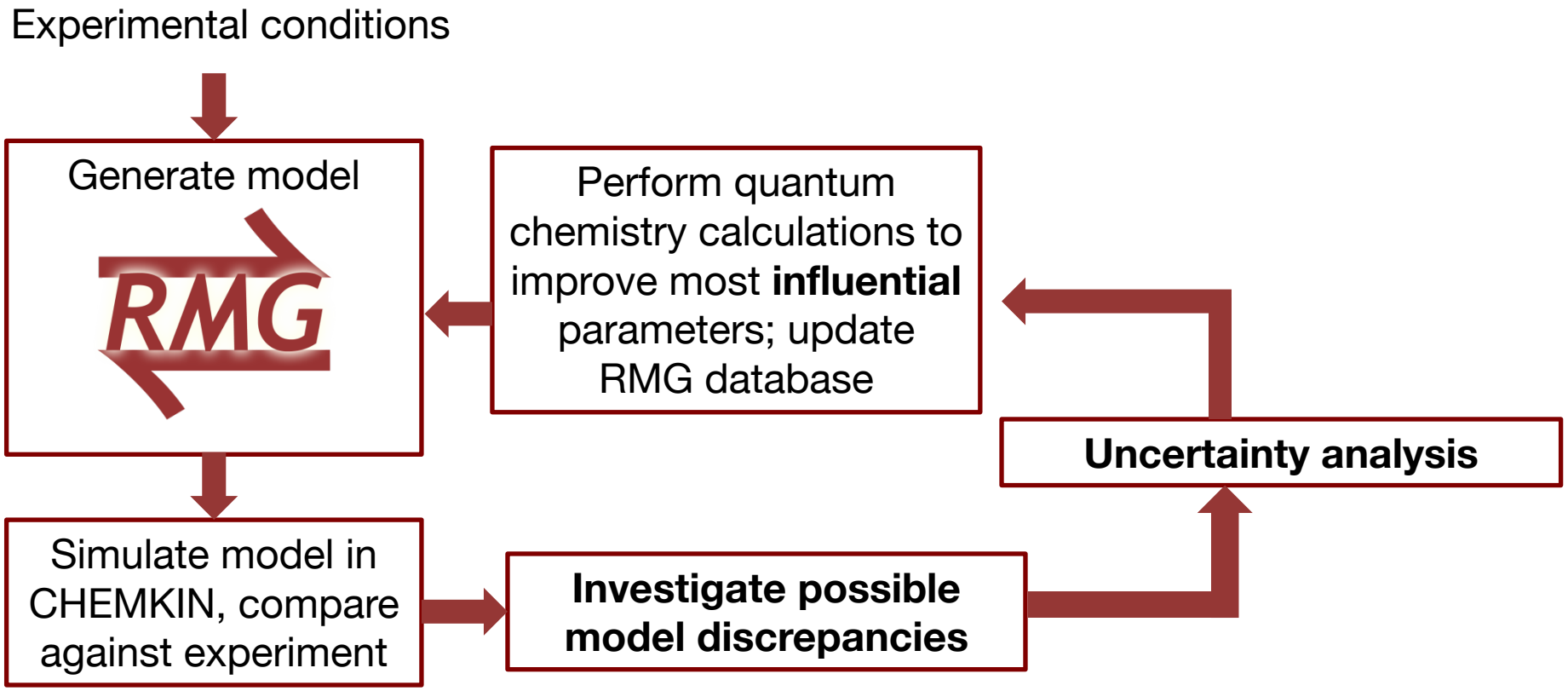
- `findParameterSourcesAndAssignUncertainies.ipynb` demonstrates what the partial uncertainty objects look like
- `localUncertainty.ipynb` demonstrates uncorrelated and correlated uncertainty propagation
- `globalUncertainty.ipynb` demonstrates uncorrelated and correlated global uncertainty propagation

Model construction workflow: old



Cycle is repeated until we have reasonable confidence in our model

Model construction workflow: new



Cycle is repeated until we have reasonable confidence in our model

Summary

Conclusions

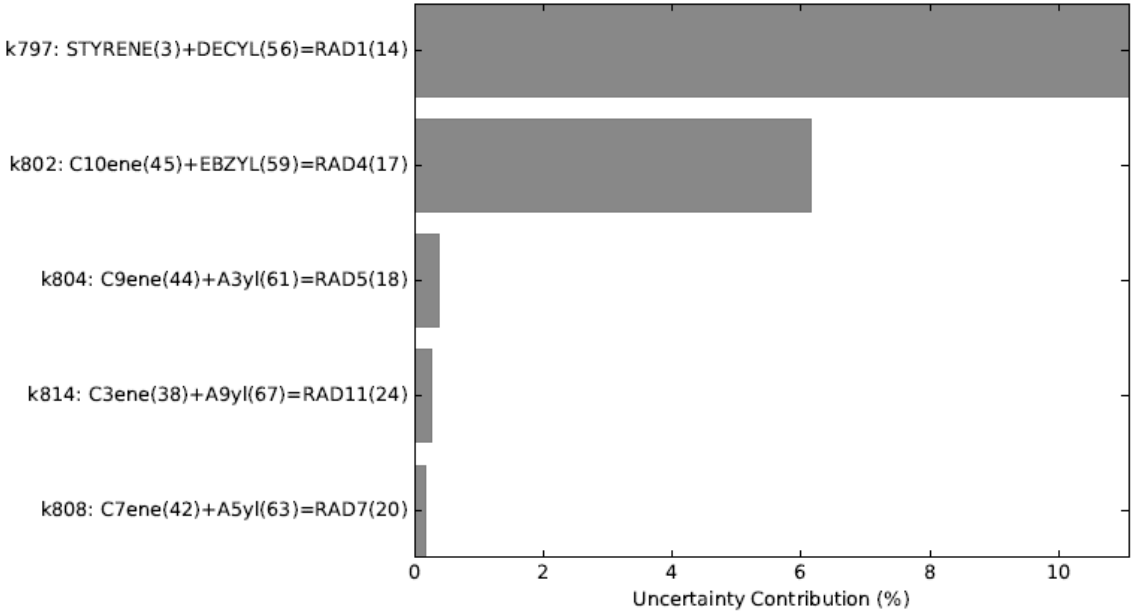
- Local uncertainties are inaccurate when parameter uncertainties are large due to the nonlinearity of chemical kinetic reaction systems
- Kineticists should consider correlations in their uncertainty analysis due to the inherent cancellation errors between groups

Local uncertainty analysis: correlated vs. independent parameters

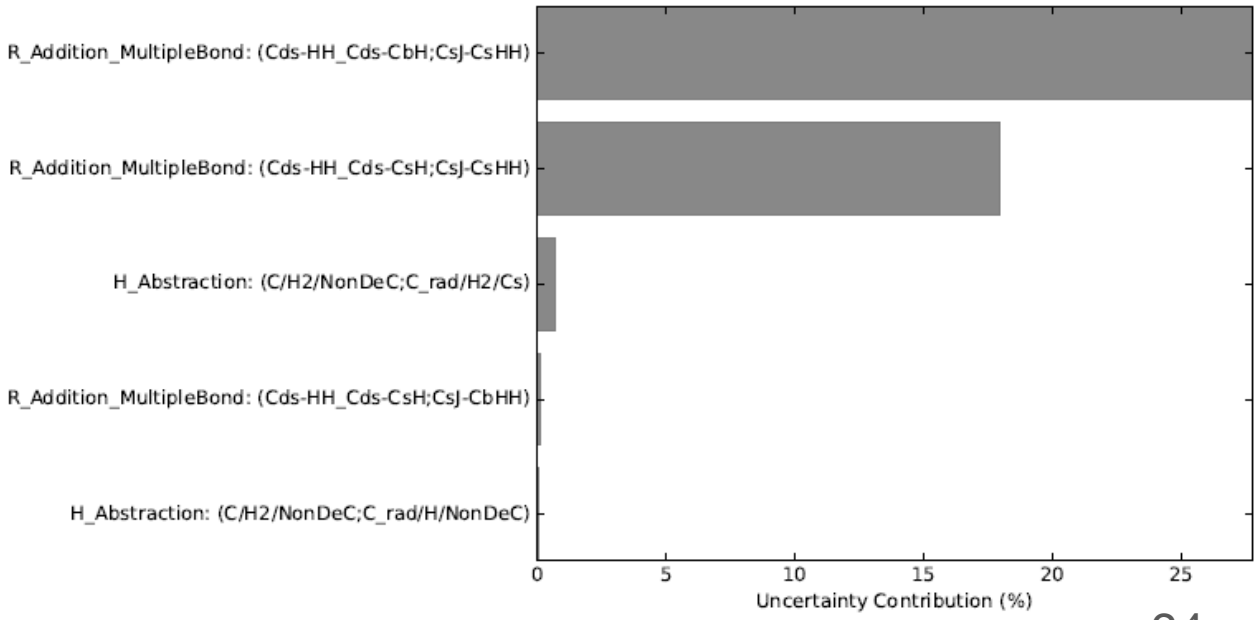
Loss of degrees of freedom…
1427 reactions collapses to **40** independent rate rules
81 species collapse to **35** independent thermodynamic parameters

	Independent parameters	Correlated parameters
Total variance $\sigma^2(\ln c)$	1.47	0.47

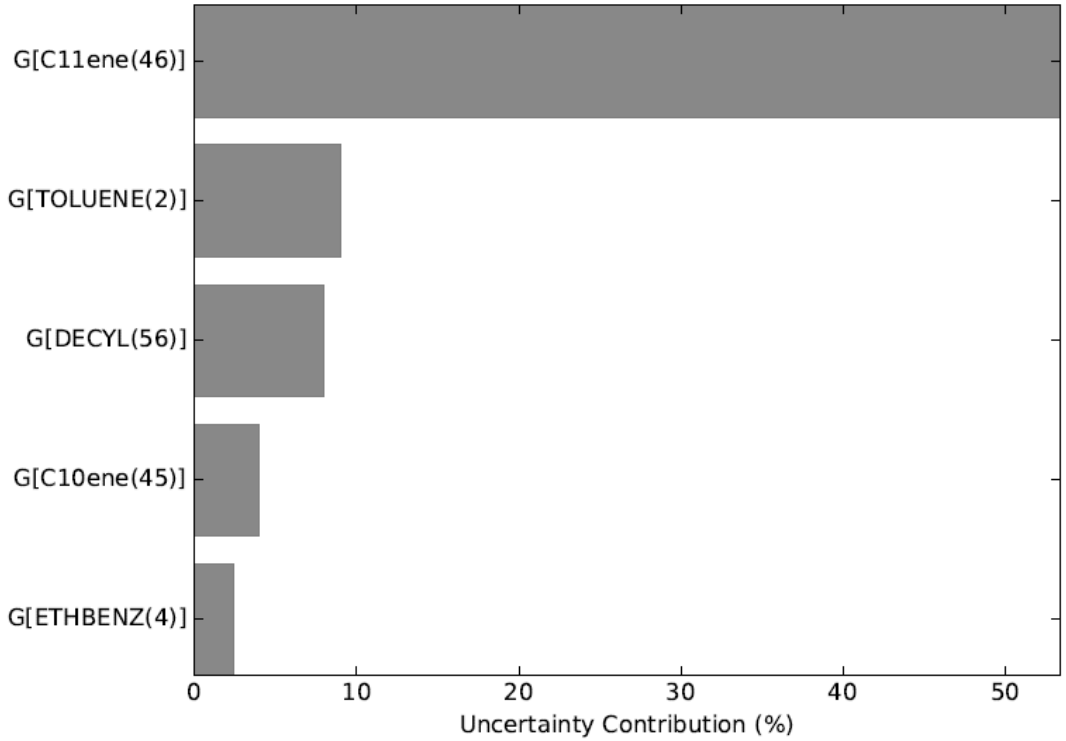
Independent Reaction Rate Coefficients

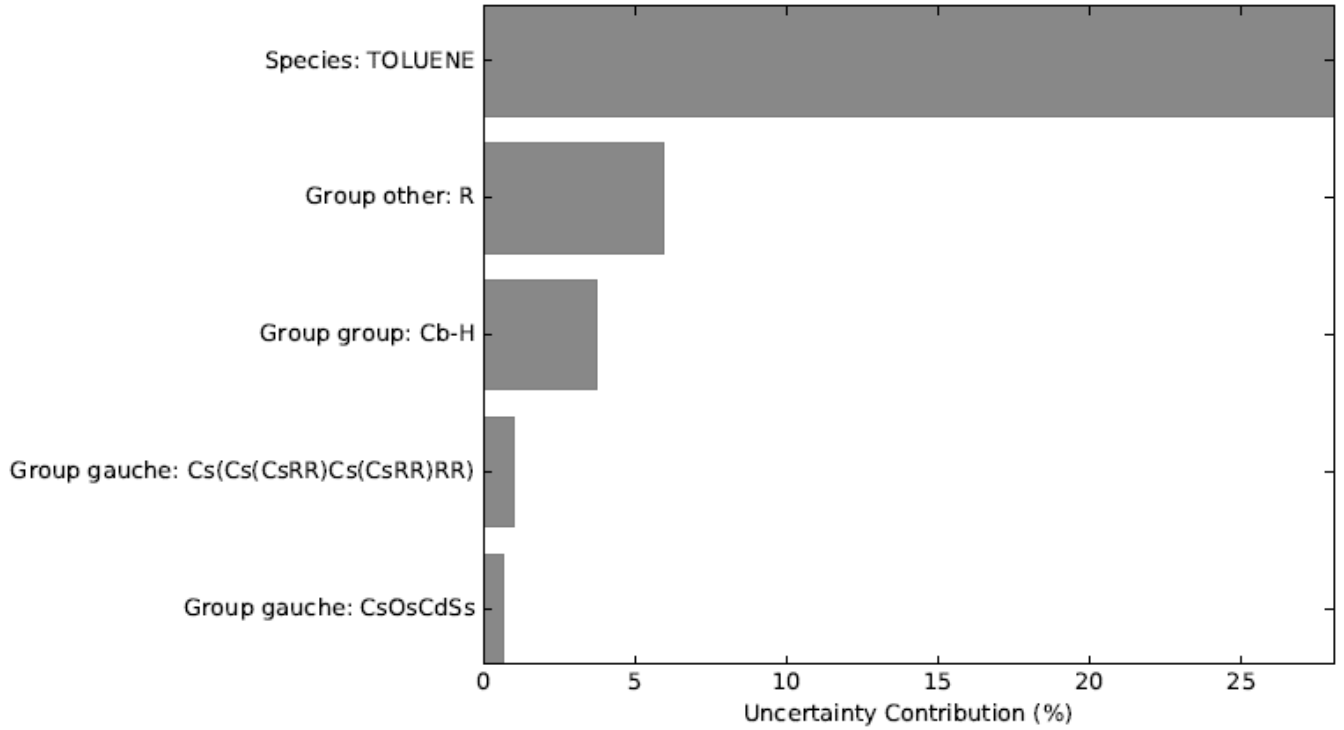


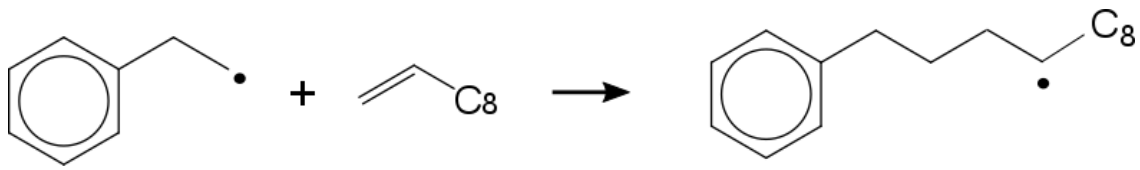
Rate Rules



Drastic reduction in uncertainty introduced by thermochemistry when group additivity values accounted for







$$\Delta G_{rxn} = \sum_{products} \Delta G^{\circ}_{f} - \sum_{reactants} \Delta G^{\circ}_{f}$$

6 **Cb-H** benzene groups

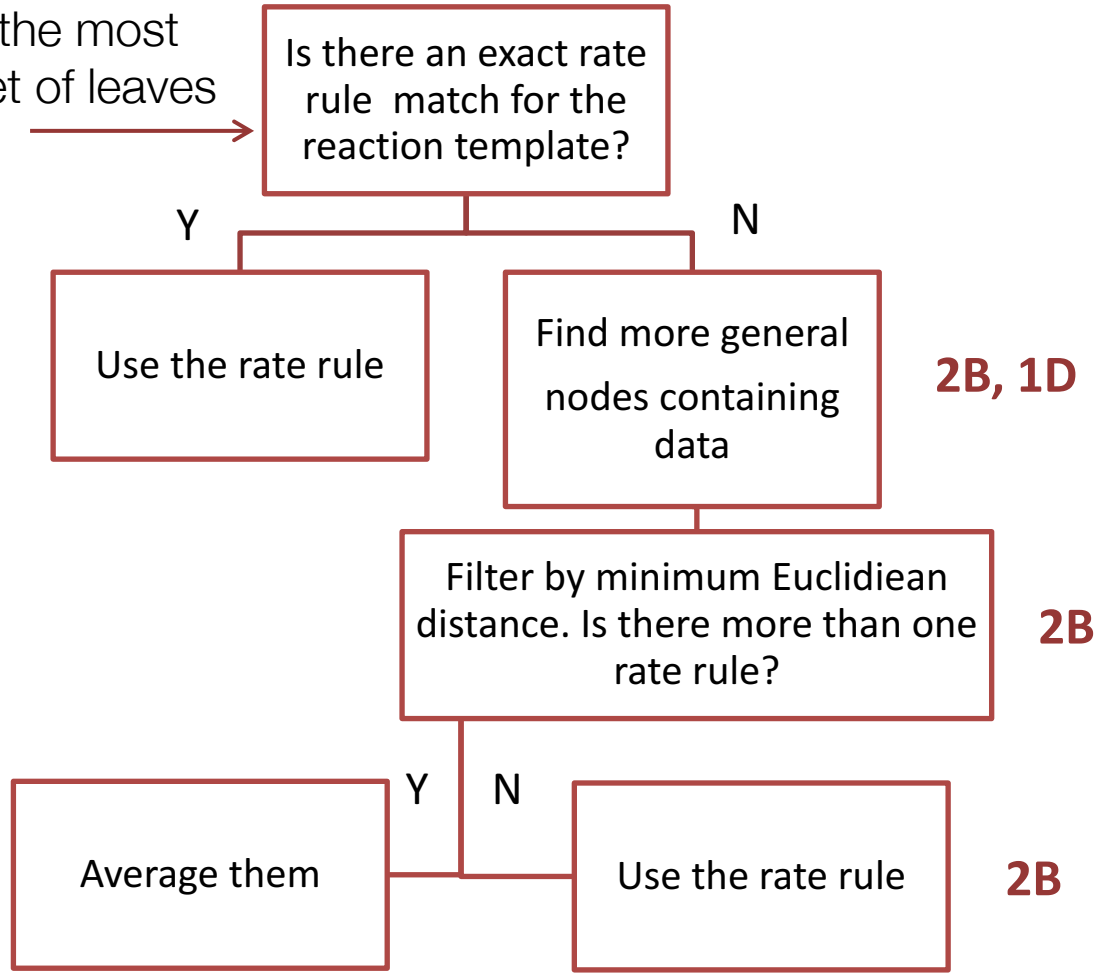
6 **Cb-H** benzene groups

Cancellation of group values reduces thermochemistry error

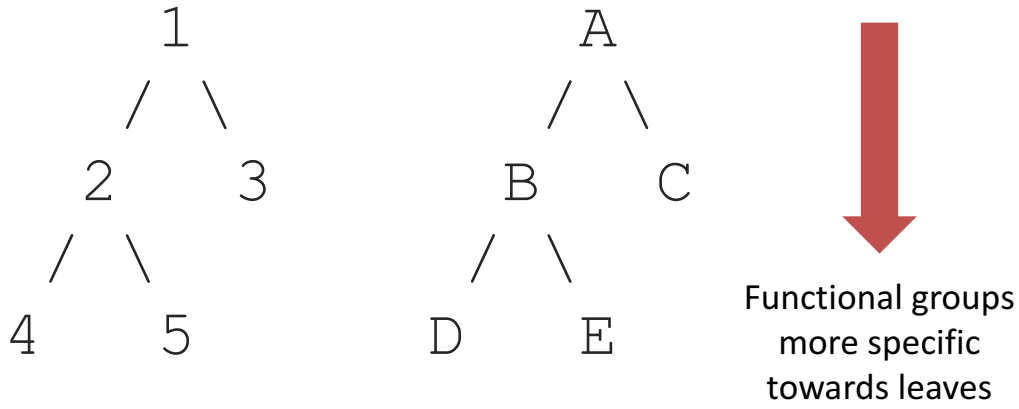
Understanding how rate rules work: new method minimizes Euclidean distance to select the best match

Identify reaction template: the most specific set of leaves in the tree

5D



XH + Yrad



Previously, we used minimum Manhattan distance
D(2B,5D) = 2, and D(1D,5D) = 2

With Euclidean distance:
D(2B,5D) = $\sqrt{1+1}$, and D(1D, 5D) = $\sqrt{0+4}$

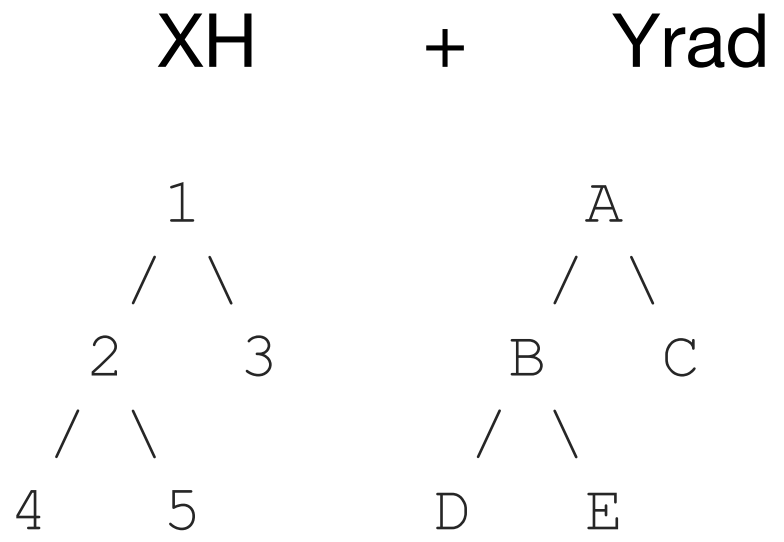
Minimizes usage of highly general nodes

Preparing the rate rule trees

1. Each family contains Training Reactions. Add these as rate rules to the most specific template in the trees.
2. Average up to fill up the trees
   - Find all cross layer template combinations i.e. 1A, 1B, 1C, 1D, 1E, etc.
   - Average the distance 1 children that exist, i.e. 1A = avg(1B, 1C, A2, A3) IF the original template contains no kinetics data

Previously:

- Did not average all cross-layer combinations → reactions tend to use more general nodes as estimates
- Used pure children averages, i.e. 1A = avg(2B, 2C, 3B, 3C) → children are not mutually exclusive and may lead to biasing of their parents



Source tracking to **original** database objects for easy investigation

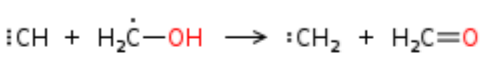
- Species source

```
prod_22(60)
```

```
{'GAV': {'radical': ['cyclopentene-4'], 'ring': ['Cyclopentene'], 'other': ['R', 'R', 'R', 'R', 'R'], 'group': ['C
s-CsCsHH', 'Cs-(Cds-Cds)CsHH', 'Cs-(Cds-Cds)CsHH', 'Cds-CdsCsH']}}
```

- Reaction source

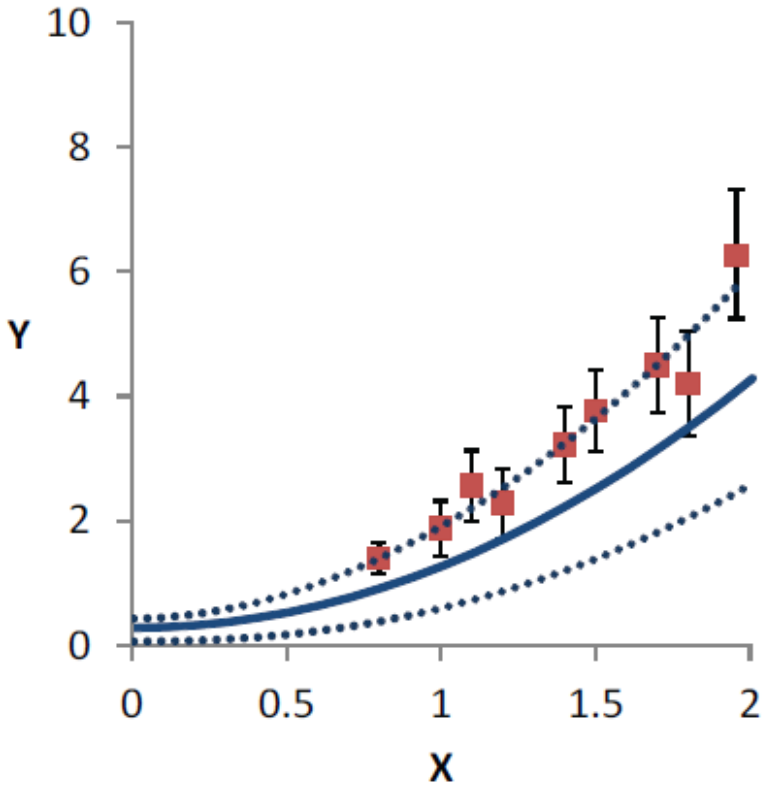
```
CH(9) + CH2OH(18) <=> CH2(11) + CH2O(15)
```



```
Original Template =  ['CH_quartet', 'O_Csrad']
Exact =  False
Rate rule sources =  ['O2b;O_Csrad', 'O_atom_triplet;O_Csrad', 'CH2_triplet;O_Csrad', 'O_pri_rad;O_Csrad', 'O_rad/
NonDeC;O_Csrad', 'O_rad/NonDeO;O_Csrad', 'Cd_pri_rad;O_Csrad', 'CO_pri_rad;O_Csrad', 'C_methyl;O_Csrad', 'C_rad/H
2/Cs;O_Csrad', 'C_rad/H2/Cd;O_Csrad', 'C_rad/H2/O;O_Csrad', 'C_rad/H/NonDeC;O_Csrad', 'C_rad/Cs3;O_Csrad', 'H_rad;
O_Csrad']
Training reaction sources =  ['C2H + CH3O <=> C2H2 + CH2O']
```

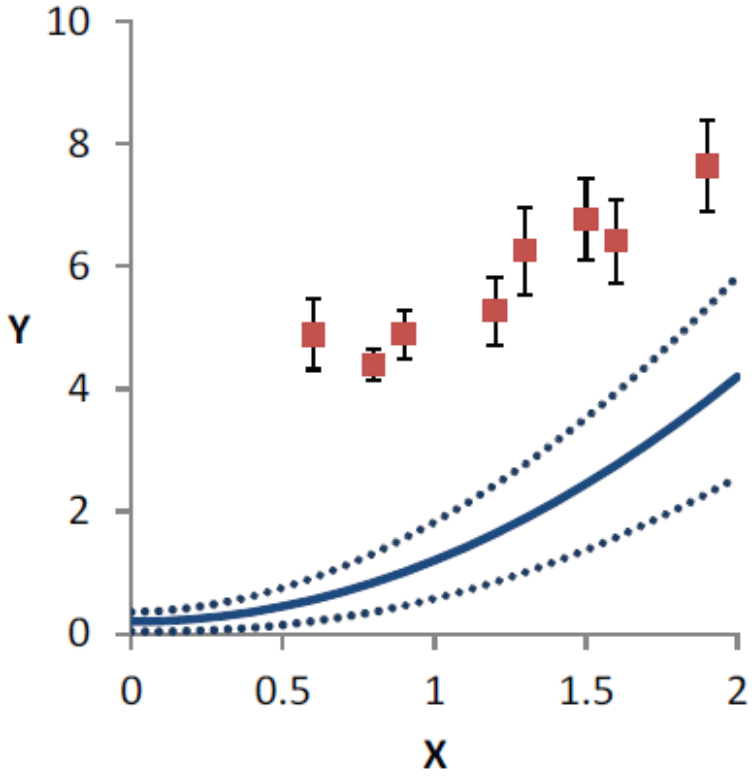
When is uncertainty analysis useful

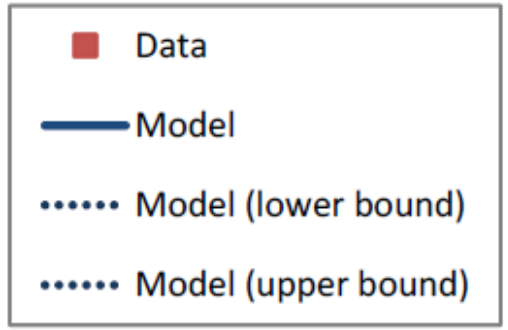
Case 1: model error bars overlap with data



Improvement in parameters improves model

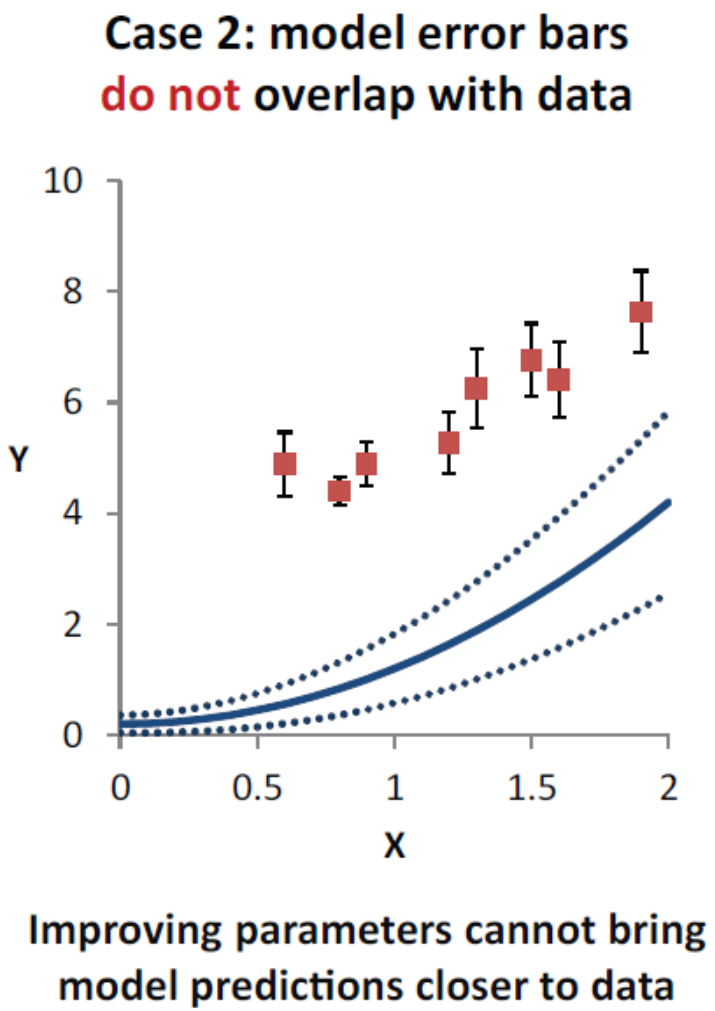
Case 2: model error bars do not overlap with data





Improving parameters cannot bring model predictions closer to data

Understanding the discrepancy between model and data



- Errors bars on input parameters in the model are underestimated
- Error bars on experimental data are underestimated
- Propagated error bounds on model due to input parameter uncertainties are underestimated
- Model structure is missing key features such as reaction paths or species
- Simulation is missing key approximations or using incorrect assumptions about experimental conditions and physics

Uncertainty analysis addresses primarily the bullet points in red