

PyFrag 2019—Automating the Exploration and Analysis of Reaction Mechanisms

Xiaobo Sun ^[a], Thomas M. Soini ^[b], Jordi Poater ^{*,[c]}, Trevor A. Hamlin ^{*,[a]} and F. Matthias Bickelhaupt ^{*,[a,d]}

We present a substantial update to the PyFrag 2008 program, which was originally designed to perform a fragment-based activation strain analysis along a provided potential energy surface. The original PyFrag 2008 workflow facilitated the characterization of reaction mechanisms in terms of the intrinsic properties, such as strain and interaction, of the reactants. The new PyFrag 2019 program has automated and reduced the time-consuming and laborious task of setting up, running, analyzing, and visualizing computational data from reaction mechanism studies to a single job. PyFrag 2019 resolves three main challenges associated with the automated computational exploration of reaction mechanisms: it (1) computes the

reaction path by carrying out multiple parallel calculations using initial coordinates provided by the user; (2) monitors the entire workflow process; and (3) tabulates and visualizes the final data in a clear way. The activation strain and canonical energy decomposition results that are generated relate the characteristics of the reaction profile in terms of intrinsic properties (strain, interaction, orbital overlaps, orbital energies, populations) of the reactant species. © 2019 The Authors. *Journal of Computational Chemistry* published by Wiley Periodicals, Inc.

DOI: 10.1002/jcc.25871

Introduction

In recent decades, significant progress has been made in the field of quantum chemistry,^[1,2] especially toward the development of density functional theory methods.^[3–5] In combination with the vast increase in computational resources now available to researchers, chemical modeling has become a modern discipline. Indeed, nowadays computational chemistry programs allow not only the modeling of a large variety of chemical systems and processes with sufficient accuracy,^[6–9] but they also provide a better, more thorough understanding of chemical phenomena, based on first principles.^[10–12] The obtained insight and understanding of a given reaction mechanism can then be used for the systematic optimization of reaction conditions and other parameters, thereby enhancing the efficiency of the chemical reaction and reducing unwanted side products.^[13,14] Computational insight, thus, greatly enhances the capabilities of the chemists and provides them with an additional and often complementary perspective to tackle chemical problems with little limitations from experimental conditions.^[15,16]

A number of theoretical models have been developed to analyze different aspects of chemical reactions and the associated changes in the chemical bonding situation. Examples are frontier molecular orbital (FMO) theory,^[17,18] Marcus theory,^[19,20] and the curve-crossing model in valence bond (VB) theory.^[21,22] An even more extensive analysis technique is the Activation Strain Model^[23] (ASM), also known as the distortion/interaction model.^[24,25] The ASM is a fragment-based approach that characterizes reactions in terms of simple and intuitive concepts, such as the effect of geometrical deformations of the reactants and various interaction terms due to changes in the electronic structure along the reaction path. For instance, we have previously used the ASM to understand how the reaction barrier varies when different bonds are activated by palladium,^[26] or how ligands can change the activating

capability of palladium,^[27] or how and why other metal centers perform differently in cross-coupling reactions compared to palladium.^[28] In addition, the ASM has been successfully applied to understand the quantitative factors governing molecular reactivity in other systems like cycloadditions,^[29–35] metalorganic catalysis,^[11,36,37] and substitution reactions.^[38–40]

The original version of PyFrag 2008^[41] allowed for the routine exploration and analysis of potential energy surfaces using the

[a] X. Sun, T. A. Hamlin, F. M. Bickelhaupt
Department of Theoretical Chemistry and Amsterdam Center for Multiscale Modeling, Vrije Universiteit Amsterdam, De Boelelaan 1083, 1081 HV, Amsterdam, Netherlands
E-mail: t.a.hamlin@vu.nl or f.m.bickelhaupt@vu.nl

[b] T. M. Soini
Software for Chemistry & Materials B.V., De Boelelaan 1083, 1081 HV, Amsterdam, Netherlands

[c] J. Poater
ICREA, Pg. Lluís Companys 23, 08010 Barcelona, Spain and Departament de Química Inorgànica i Orgànica & IQTCUB, Universitat de Barcelona, 08028, Barcelona, Catalonia, Spain
E-mail: jordi.poater@ub.edu

[d] F. M. Bickelhaupt
Institute for Molecules and Materials, Radboud University, Heyendaalseweg 135, 6525 AJ, Nijmegen, Netherlands
E-mail: f.m.bickelhaupt@vu.nl

Contract Grant sponsor: China Scholarship Council; Contract Grant sponsor: Generalitat de Catalunya; Contract Grant number: 2017SGR348; Contract Grant sponsor: Nederlandse Organisatie voor Wetenschappelijk Onderzoek; Contract Grant sponsor: Spanish MINECO; Contract Grant numbers: CTQ2016-77558-R, MDM-2017-0767; Contract Grant sponsor: Ministerio de Ciencia, Innovación y Universidades; Contract Grant numbers: CTQ2016-77558-R, MDM-2017-0767

This is an open access article under the terms of the Creative Commons Attribution-NonCommercial License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited and is not used for commercial purposes.

© 2019 The Authors. *Journal of Computational Chemistry* published by Wiley Periodicals, Inc.

ASM. That version consisted of a driver invoking the Amsterdam Density Functional (ADF) software,^[42–44] which in turn made use of the concept of molecular fragments for the analysis of chemical bonds.^[12] After many successful applications, we have now overhauled the original PyFrag 2008 code with the aim to (1) make extensive usage of the scripting framework provided by the Python Library for Automating Molecular Simulations (PLAMS)^[45] and QMflows,^[46] thereby enabling PyFrag 2019 to (2) be compatible with other quantum chemistry programs such as Gaussian,^[47] Orca,^[48] and Turbomole^[49] and their respective analysis tools; and (3) completely automate the workflow of the ASM approach. In order to monitor the computational progress at runtime, PyFrag 2019 uses Bokeh,^[50] an interactive visualization Python library, to generate an html file summarizing the computational results in tables, figures, and videos.

PyFrag 2019 requires the user to simply supply approximate geometries of the stationary points (reactants, transition states, and products). PyFrag 2019 then sets up and runs the optimizations of the stationary points and also the complex workflows associated with the ASM approach involving many calls to computational chemistry programs in sequential order or, where possible, in parallel. The computational progress and the convergence of the required geometry and transition state optimizations can be monitored at runtime. After completing the computations, PyFrag 2019 then visualizes the results in a clear, automated, and informative way.

The remainder of this article is organized as follows: the subsequent section describes the ASM and energy decomposition analysis (EDA) method, which are used to analyze properties and quantities along the reaction path. Afterward, we provide specific examples of the complete workflow as implemented in PyFrag 2019 being employed to understand the reactivity of, among many others tested, three widely studied reactions: the oxidative addition of C–X bonds by iron catalyst,^[28] 1,3-dipolar cycloaddition reactions,^[29] and the analysis of the nucleophilicity and leaving-group ability of backside S_N2 reactions.^[38]

Methods

The ASM is a fragment-based approach used to understand the character of the reaction mechanism and to gain insights into the overall reaction energy profile. Usually, the reaction profile is determined by the interplay between two reactants or fragments. The bonding energy ΔE is decomposed along the intrinsic reaction coordinate (IRC)^[51,52] into the strain energy ΔE_{strain} that is associated with the geometrical deformation of the individual reactants as the process takes place, plus the actual interaction energy ΔE_{int} between the deformed reactants (see eq. 1).

$$\Delta E = \Delta E_{\text{strain}} + \Delta E_{\text{int}} \quad (1)$$

The interaction energy ΔE_{int} between the deformed reactants is further analyzed in the conceptual framework provided by the Kohn-Sham molecular orbital (KS-MO) model, using a quantitative EDA scheme (eq. 2).^[12,53–55] While any of the aforementioned computational chemistry programs can be used to perform the

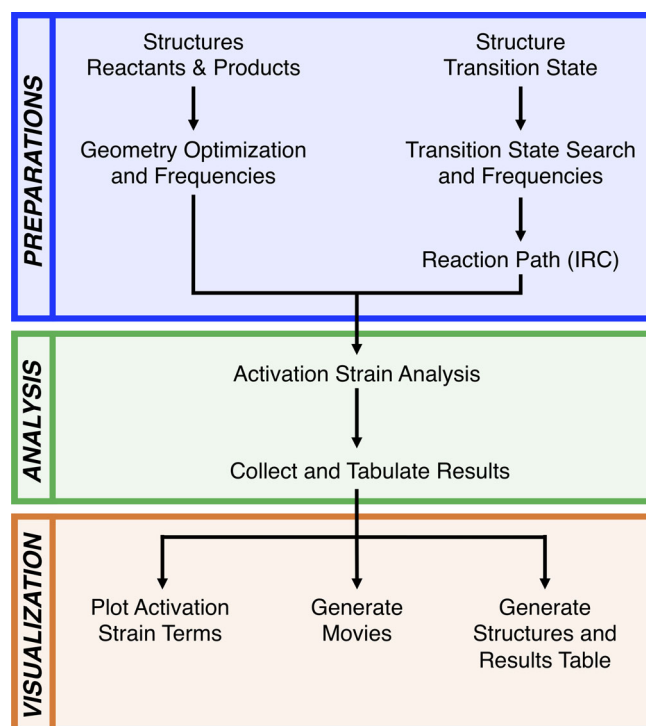


Figure 1. Schematic flowchart of the PyFrag 2019 program, including the three steps of (i) preparations, (ii) analysis, and (iii) visualization. [Color figure can be viewed at wileyonlinelibrary.com]

activation strain analysis, it should be noted that the EDA presented in this article is a unique feature of ADF.^[42–44]

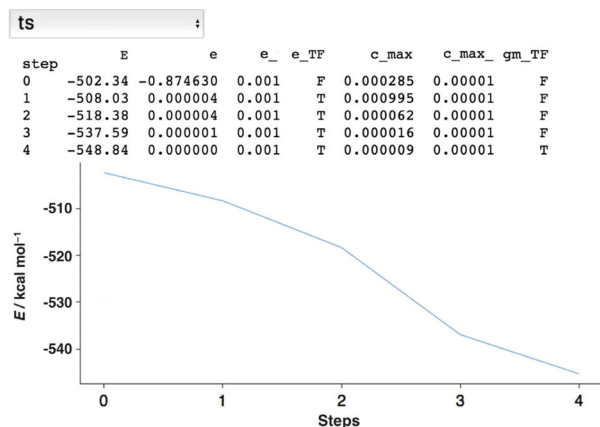
$$\Delta E_{\text{int}} = \Delta V_{\text{elstat}} + \Delta E_{\text{Pauli}} + \Delta E_{\text{oi}} \quad (2)$$

The ΔV_{elstat} energy corresponds to the classical Coulomb interaction between the unperturbed charge distributions of the deformed reactants and is usually attractive. The Pauli repulsion energy ΔE_{Pauli} comprises the destabilizing interactions between occupied orbitals (more precisely, between same-spin orbitals) on the respective reactants and is responsible for steric repulsion. The orbital interaction energy ΔE_{oi} accounts for charge transfer (interaction between occupied orbitals on one fragment with unoccupied orbitals on the other fragment, including the HOMO–LUMO interactions) and polarization (empty-occupied orbital mixing on one fragment due to the presence of another fragment).

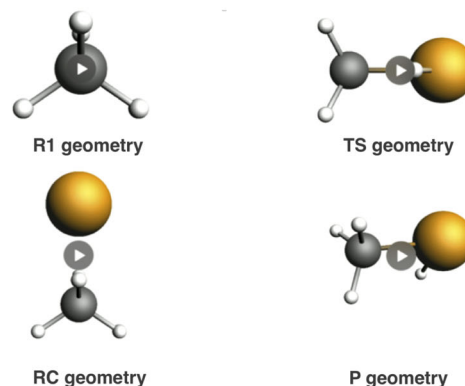
Description of the Program

An overview of the complete PyFrag 2019 workflow is shown in Figure 1. Our new implementation consists of three parts: calculation and preparation of the whole reaction path, activation strain analysis based on this computed reaction path, and finally the tabulation and visualization of all results. The user only needs to provide basic information like initial guesses for the molecular structures and the specific parameters required for the DFT calculations (see the Supporting Information for an example input file). PyFrag 2019 then automatically submits the necessary computational jobs to (eventually remote) computing nodes and gathers the corresponding results after the completion of each

(a) Current Job Status



(b) Movies of Job Optimizations



(c) Summary of latest coordinates and other information

Geometry (cartesian) and Energy (kcal/mol)	
R1.xyz,	-554.0858
1.C	0.00000000 0.00000000 0.00000000
2.H	0.63276400 -0.63276400 0.63276400
3.H	-0.63276400 -0.63276400 -0.63276400
4.H	-0.63276400 0.63276400 0.63276400
5.H	0.63276400 0.63276400 -0.63276400
P.xyz,	-552.4892
1.C	-2.10134690 -2.41901732 0.18620613
2.Pd	-2.73145901 -0.57025833 0.41934115

(d) Activation Strain Analysis

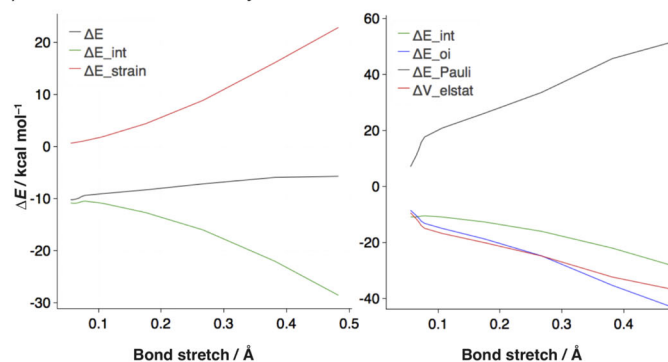


Figure 2. Snapshots from the html page generated by PyFrag 2019 that shows (a) the current job status (Transition State); (b) corresponding movies of each geometrical optimization process; (c) the summary of the geometries and energies of stationary points on the potential energy surface [reactant (R1), reactant complex (RC), transition state (TS), and product (P)]; and (d) after completion of the workflow, the results of the activation strain analysis. For the current job status, various types of information are provided, including: a plot of energy (E) against each step, convergence information like energy change (e), constrained gradient max (c_max), their responding converge standard (e_c, c_max), as well as if they are converged (e_{TF}, gm_{TF}). The user can click the box and select other stationary points, such as the reactant or product and obtain relevant data. [Color figure can be viewed at wileyonlinelibrary.com]

job. The structures of the reactants and products of the reaction are optimized first, followed by a search for the transition state starting from a reasonable initial guess for its geometry. A frequency analysis is then performed to ensure that the previously optimized structures and the transition state are indeed stationary points on the potential energy surface. A deeper understanding of a reaction mechanism can often be gained only by inspecting the entire reaction path rather than the stationary points alone.^[56,57] For this reason, a reaction path between the transition state and the initial and final structures, respectively, is then generated by means of an IRC calculation.^[51,52] The activation strain analysis is then performed to identify the main mechanistic characteristics of a given reaction. During the execution of the workflow, relevant information about the current status is collected and written into an html output file, an example of which is depicted in Figure 2a-c. Upon completion of the workflow, the activation strain diagram is generated (Fig. 2d). The activation strain analysis results are also collected and stored as a text file, which can be exported and plotted in MS Excel either manually or automatically with the help of a supplementary function in PyFrag 2019. In cases where the reaction path is already available, PyFrag 2019 is also able to import the

coordinates and directly proceed with the activation strain analysis workflow using either ADF,^[42] Gaussian,^[47] ORCA,^[48] or Turbomole.^[49] ExcelAutomat 1.3^[58] (compatible with Gaussian^[47] and GAMESS-US^[59-61]) and autoDIAS^[62] (compatible with Gaussian,^[47] ORCA,^[48] NWChem,^[63] and Q-Chem^[64]) have recently been published and have been shown to be useful in automating the activation strain analysis along a provided reaction coordinate. Note, however, that while making advances toward an automated process, both of these programs are incompatible with ADF and thus are unable to perform the EDA.^[12,53-55] We should also note that the present version of PyFrag 2019 is not yet compatible with open-shell calculations.

To facilitate the rapid evaluation of the workflow and to display the status of the entire computation, PyFrag 2019 provides the functionality to periodically extract relevant data from the running calculation and visualize it as an html output (see Fig. 2). A continuously updated video can be generated to examine the reaction path in real time. In the case of a failure of a computation, such as geometry optimization, the workflow will continue carrying out all other computations, unless they explicitly depend on the job that has failed. When such erratic behavior occurs, the user can modify the input file, which in turn will be

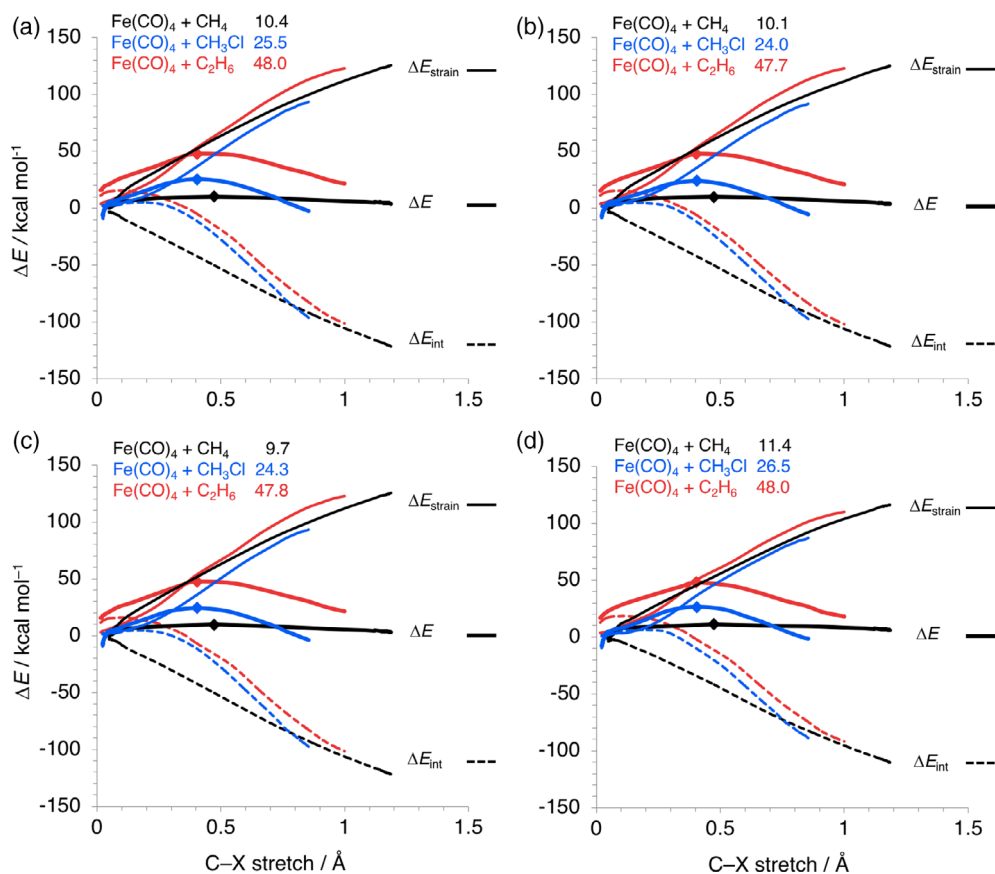


Figure 3. Activation strain analysis for the oxidative addition of CH₃-X bonds (X=H, Cl, CH₃; black, blue, red, respectively) to singlet Fe(CO)₄ computed at (a) ZORA-OPBE/TZ2P level with ADF 2017; (b) DKH-OPBE/6-311++G(d,p) level with Gaussian 09; (c) ZORA-OPBE/ZORA-def2-TZVP SARC/J level with ORCA 4.0; and (d) DKH-BLYP/TZVP level with Turbomole 6.5. Energy barriers relative to the reactants for the transition state are included (see upper left corner) and the positions of TS are marked by diamonds. [Color figure can be viewed at wileyonlinelibrary.com]

detected and accounted for by PyFrag 2019 and used to restart any affected computational jobs automatically.

Results and Discussion

To demonstrate the ability of PyFrag 2019 to automate the steps necessary for the mechanistic analysis summarized in Figure 1, we proceed by presenting three application cases involving different chemical reactions: an (1) oxidative addition to an iron catalyst, a (2) series of 1,3-dipolar cycloadditions, and (3) bimolecular nucleophilic substitution (S_N2) reactions. In order to perform these analyses, as mentioned above, the user must only supply approximate geometries of the stationary points (reactants, transition state, and products). After completion of the calculation, the user can then plot the activation strain and EDA that were generated by PyFrag 2019 with the selected quantum chemistry program. In this section, we limit our discussion to the final step involving the visualization step, since this step of the workflow is the most informative for understanding the factors governing the reactivity of the specific reactions and to check the correct performance of PyFrag 2019.

Reaction 1: Oxidative addition by an iron catalyst

Catalytic reactions are a key transformation in modern synthetic chemistry^[65,66] and the oxidative addition step is generally the first and rate-determining step in most catalytic cycles.^[67,68] Our first example involves the C-X bond activation via oxidative addition of CH₃X substrates (X = H, Cl, CH₃) to a model iron catalyst Fe(CO)₄ with singlet spin state.^[28] The results from the

activation strain analysis carried out with PyFrag 2019 using ADF, Gaussian, ORCA, and Turbomole as computational engines are shown in Figure 3a-d. As expected, but still important, we find comparable results and the same trends in reactivity for all programs employed, which confirms these individual implementations.

The activation barrier (ΔE) depends on two chemically intuitive terms, namely the strain (ΔE_{strain}) and interaction (ΔE_{int}) energies (see eq. 1). We focus the following discussion on the results obtained with ADF as summarized in Figure 3a. The energy barriers for the bond activation of C-H, C-Cl, and C-C by Fe(CO)₄ increase from 10.4 to 25.5 to 48.0 kcal mol⁻¹, respectively. A weaker catalyst-substrate interaction results in a higher barrier for the C-Cl activation compared to the C-H activation. The C-C bond activation exhibits the highest barrier due to a highly destabilizing activation strain and a weakly stabilizing interaction energy compared to the activation of the C-H and the C-Cl bonds. The strength of the ΔE_{int} for these bond activation processes was traced back to the nature of the σ and σ^* orbitals of the C-H, C-Cl or C-C bonds. In particular, the $\sigma_{\text{C-H}}$ lacks a nodal plane and has a significant overlap with the iron catalyst $3d_{\sigma}$ orbital in a σ -donating manner. At variance, the $\sigma_{\text{C-Cl}}$ and $\sigma_{\text{C-C}}$ for C-Cl and C-C, respectively, have a $2p$ -nodal plane that results in the cancellation of orbital overlap with the iron $3d_{\sigma}$. These orbital overlap values are easily printed by PyFrag 2019 in the same manner as the activation strain terms. Finally, the more destabilizing ΔE_{strain} for C-C activation compared to C-Cl activation simply originates from the higher strength of the C-C bond.^[69]

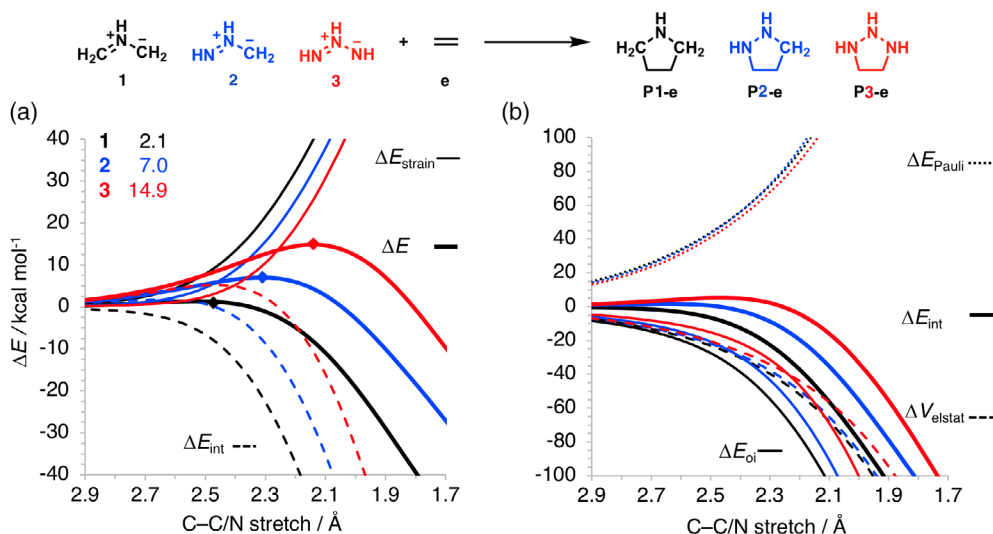


Figure 4. (a) Activation strain analyses and (b) energy decomposition analyses of the cycloaddition reactions between 1,3-dipoles 1–3 and ethylene (e), computed at the BP86/TZ2P level. Energy barriers relative to the reactants for the transition state are included (see upper left corner), and the positions of TS are marked by diamonds. [Color figure can be viewed at wileyonlinelibrary.com]

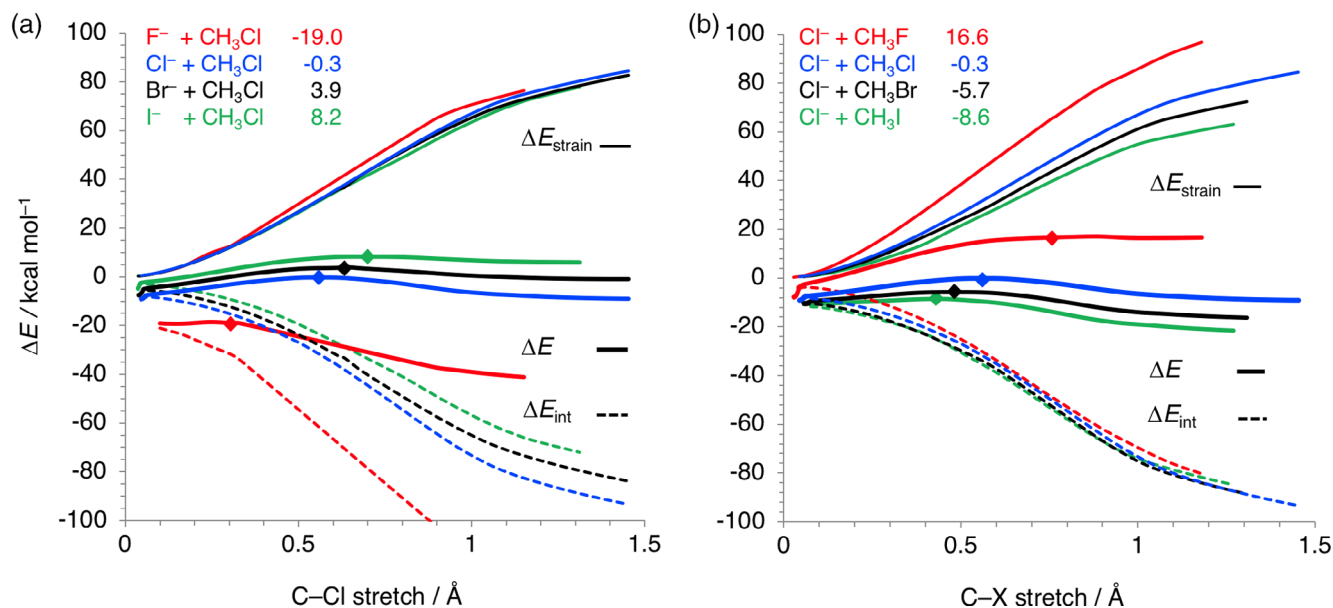


Figure 5. Activation strain analysis of the S_N2 reactions for (a) $X^- + \text{CH}_3\text{Cl}$ ($X^- = \text{F}^-, \text{Cl}^-, \text{Br}^-$, and I^-) and (b) $\text{Cl}^- + \text{CH}_3\text{Y}$ ($\text{Y} = \text{F}, \text{Cl}, \text{Br}$, and I), computed at the OLYP/TZ2P level. Energy barriers relative to the reactants for the transition state are also included (see upper left corner), and the positions of TS are marked by diamonds. [Color figure can be viewed at wileyonlinelibrary.com]

Reaction 2: 1,3-dipolar cycloaddition reactions

Our second example investigates the 1,3-dipolar cycloaddition (1,3-DCA) reactivities of a series of aza-1,3-dipoles 1 (black), 2 (blue), 3 (red) with ethylene (e) as shown in Figure 4. 1,3-DCA cycloadditions are important reactions in heterocyclic chemistry,^[70] materials chemistry,^[71–73] and chemical biology.^[74–76] The cycloaddition barrier increases from 1 to 2 to 3 corresponding to a sharp decrease in reactivity. Our activation strain analysis (Fig. 4a) reveals that the differences in the barrier heights for these 1,3-DCA reactions originate from differences in ΔE_{int} and not from ΔE_{strain} , as originally proposed.^[25,77,78] Despite the reaction of 1 having the most destabilizing ΔE_{strain} it goes with the lowest barrier due to a highly stabilizing ΔE_{int} . Decomposition of the interaction energy reveals that the trends in the Pauli repulsion, ΔE_{Pauli} , are more or less offset by the trends in the electrostatic interaction, ΔV_{elstat} (Fig. 4b). The

orbital interactions, ΔE_{oi} , on the other hand, play a dominant role in determining the trend in interaction energy, and thus, the height of the reaction barriers. The orbital interaction becomes weaker from 1 to 2 to 3 due to a larger, less favorable FMO energy gaps. In addition, as the number of nitrogen atoms in the 1,3-dipole increases from 1 to 3 for compounds 1, 2, and 3, respectively, the FMO overlap decreases, due to the more contracted 2p-orbitals of nitrogen compared to those of carbon.^[79] These two effects, the larger FMO energy gaps and a decrease in FMO overlap, reinforce each other and result in a less stabilizing orbital interaction curve, and thus a higher and less favorable activation barrier.

Reaction 3: Bimolecular nucleophilic substitution (S_N2)

Our third example involves the textbook bimolecular nucleophilic substitution (S_N2) reaction.^[80] We have analyzed the nucleophilicity

and leaving-group ability of halogens in backside S_N2 reactions for $X^- + CH_3Cl$ ($X^- = F^-, Cl^-, Br^-,$ and I^-) and $Cl^- + CH_3Y$ ($Y = F, Cl, Br,$ and I), respectively.^[38] In Figure 5a, we show that for a given leaving group, the S_N2 barrier increases as the nucleophile is varied from F^- to I^- . This trend stems entirely from the ΔE_{int} curves, as the ΔE_{strain} curves are nearly identical and superimposed. The nucleophilicity of these halides is determined by the electron-donor capability of the nucleophile. Specifically, the HOMO–LUMO interaction becomes less stabilizing as the nucleophile is varied along $X^- = F^-, Cl^-, Br^-,$ and I^- and this, in turn, leads to a systematically less stabilizing interaction. This can be understood in terms of the orbital energy of the X^- HOMO, which is the highest (most destabilized) for F^- and decreases (becomes stabilized) to I^- , thus leading to a larger HOMO–LUMO gap.

For a given nucleophile, the S_N2 barrier decreases as the leaving group is varied from F to I for the $Cl^- + CH_3Y$ ($Y = F, Cl, Br,$ and I) reactions. Interestingly, in this case, the reactivity trend is entirely determined by the ΔE_{strain} curves, as the ΔE_{int} curves are nearly identical for the different reactions. The ΔE_{strain} curves become less destabilizing as the leaving group is varied from $Y = F, Cl, Br,$ and I . The leaving group ability is directly related to the trend in C–Y bond strengths, with C–F being the strongest bond ($\Delta H_{BDE} = 113.7 \text{ kcal mol}^{-1}$) and C–I the weakest one ($\Delta H_{BDE} = 60.7 \text{ kcal mol}^{-1}$).^[81,82]

Conclusions

The PyFrag 2019 program is presented, featuring significant improvements over its predecessor,^[41] with the aim to facilitate the computational exploration and analysis of reaction mechanisms in an automated fashion. In particular, PyFrag 2019 resolves three main challenges associated with computing the potential energy surface of a particular reaction: it now (1) computes the reaction path by carrying out multiple parallel calculations using initial coordinates provided by the user; (2) monitors the entire workflow process; and (3) tabulates and visualizes the final data in a clear way. The automated activation strain analysis in PyFrag 2019 is compatible with many quantum chemical software packages, including ADF^[42], Gaussian^[47], Orca,^[48] and Turbomole.^[49]

PyFrag 2019 requires the user to only provide approximate geometries for the respective stationary points on the potential energy surface (reactants, transition states, products, and intermediates). The steps of the reaction path workflow involve first running the geometry optimizations of stationary points using the supplied coordinates followed by an IRC calculation to generate the potential energy surface. Next, an activation strain and canonical EDA (when using ADF) is performed. The final step of the workflow involves the visualization (plotting/tabulation) of the results. These activation strain and energy decomposition results generated from PyFrag 2019 allow the user to relate the characteristics of the reaction profile in terms of intrinsic properties (strain, interaction, orbital overlaps, orbital energies, populations) of the reactant species. PyFrag 2019 has been tested, among many others, on three widely studied chemical reactions, including the oxidative addition of C–X bonds by an iron catalyst, the reactivity of 1,3-dipolar cycloaddition reactions, and the analysis of the nucleophilicity and leaving-group ability

of S_N2 reactions. The novel workflow implemented in PyFrag 2019 has been extensively parallelized to make efficient usage of the available computational resources.


We expect PyFrag 2019 to facilitate the systematic analysis of mechanistic features involving the screening, with detailed analyses, of large amounts of potential reaction candidates. To improve the efficiency of such screening workflows even further, also additional quantum chemistry codes such as DFTB^[83] or GFN-xTB^[84] can be employed. The open source PyFrag 2019 code can be retrieved online along with an introductory tutorial.^[85,86]

Acknowledgments

We thank the China Scholarship Council (CSC), the Spanish MIN-ECO (CTQ2016-77558-R and MDM-2017-0767), the Generalitat de Catalunya (2017SGR348), and the Netherlands Organization for Scientific Research (NWO) for financial support.

Keywords: density functional calculations · automation · reaction mechanisms · activation strain model · energy decomposition analysis · program

How to cite this article: X. Sun, T. M. Soini, J. Poater, T. A. Hamlin, F. M. Bickelhaupt. *J. Comput. Chem* **2019**, *40*, 2227–2233. DOI: 10.1002/jcc.25871

 Additional Supporting Information may be found in the online version of this article.

- [1] T. A. Albright, J. K. Burdett, M. H. Whangbo, *Orbital Interactions in Chemistry*, 2nd ed., John Wiley & Sons, Hoboken, NJ, **2013**.
- [2] L. Piela, *Ideas of Quantum Chemistry*, Elsevier, Amsterdam, NL, **2013**.
- [3] A. D. Becke, *J. Chem. Phys.* **2014**, *140*, 18A301.
- [4] L. Goerigk, S. Grimme, *J. Chem. Theory Comput.* **2011**, *7*, 291.
- [5] Y. Zhao, D. G. Truhlar, *Theor. Chem. Acc.* **2008**, *120*, 215.
- [6] J. P. Perdew, L. A. Constantin, *Phys. Rev. B* **2007**, *75*, 155109.
- [7] A. D. Becke, *J. Chem. Phys.* **2013**, *138*, 074109.
- [8] Y. Zhao, N. E. Schultz, D. G. Truhlar, *J. Chem. Phys.* **2005**, *123*, 161103.
- [9] Y. Zhao, D. G. Truhlar, *Acc. Chem. Res.* **2008**, *41*, 157.
- [10] E. J. Baerends, O. V. Gritsenko, R. van Leeuwen, In *Chemical Applications in Density Functional Theory*; B. B. Laird, R. B. Ross, T. Ziegler, Eds., American Chemical Society, Washington, DC, **1996**, p. 20.
- [11] A. G. Green, P. Liu, C. A. Merlic, K. N. Houk, *J. Am. Chem. Soc.* **2014**, *136*, 4575.
- [12] F. M. Bickelhaupt, E. J. Baerends, In *Reviews in Computational Chemistry*, Vol. 15; K. B. Lipkowitz, D. B. Boyd, Eds., Wiley-VCH, New York, **2000**.
- [13] T. A. Hamlin, B. J. Levandowski, A. K. Narsaria, K. N. Houk, F. M. Bickelhaupt, *Chem. Eur. J.* **2019**, *25*, 6342.
- [14] L. P. Wolters, W. J. van Zeist, F. M. Bickelhaupt, *Chem. Eur. J.* **2014**, *20*, 11370.
- [15] M. A. van Bochove, G. Roos, C. Fonseca Guerra, T. A. Hamlin, F. M. Bickelhaupt, *Chem. Commun.* **2018**, *54*, 3448.
- [16] A. K. Narsaria, J. Poater, C. Fonseca Guerra, A. W. Ehlers, K. Lammertsma, F. M. Bickelhaupt, *J. Comput. Chem.* **2018**, *39*, 2690.
- [17] R. Hoffmann, *Angew. Chem. Int. Ed. Engl.* **1982**, *21*, 711.
- [18] K. Fukui, *Angew. Chem. Int. Ed. Engl.* **1982**, *21*, 801.
- [19] R. A. Marcus, *J. Chem. Phys.* **1956**, *24*, 966.
- [20] R. A. Marcus, *J. Chem. Phys.* **1956**, *24*, 979.
- [21] A. Pross, S. Shaik, *Acc. Chem. Res.* **1983**, *16*, 363.
- [22] D. Usharani, D. Janardanan, C. Li, S. Shaik, *Acc. Chem. Res.* **2013**, *46*, 471.
- [23] F. M. Bickelhaupt, K. N. Houk, *Angew. Chem. Int. Ed.* **2017**, *56*, 10070.
- [24] D. H. Ess, K. N. Houk, *J. Am. Chem. Soc.* **2007**, *129*, 10646.
- [25] D. H. Ess, K. N. Houk, *J. Am. Chem. Soc.* **2008**, *130*, 10187.
- [26] P. Vermeeren, X. Sun, F. M. Bickelhaupt, *Sci. Rep.* **2018**, *8*, 10729.

- [27] W. J. van Zeist, R. Visser, F. M. Bickelhaupt, *Chem. Eur. J.* **2009**, *15*, 6112.
- [28] X. Sun, M. V. J. Rocha, T. A. Hamlin, J. Poater, F. M. Bickelhaupt, *Phys. Chem. Chem. Phys.* **2019**, *21*, 9651.
- [29] T. A. Hamlin, D. Svatunek, S. Yu, L. Ridder, I. Infante, L. Visscher, F. M. Bickelhaupt, *Eur. J. Org. Chem.* **2019**, 378.
- [30] S. Liu, Y. Lei, X. Qi, Y. Lan, *J. Phys. Chem. A* **2014**, *118*, 2638.
- [31] R. Jin, S. Liu, Y. Lan, *RSC Adv.* **2015**, *5*, 61426.
- [32] Y. García-Rodeja, I. Fernández, *J. Org. Chem.* **2017**, *82*, 8157.
- [33] Y. García-Rodeja, M. Solà, I. Fernández, *Phys. Chem. Chem. Phys.* **2018**, *20*, 28011.
- [34] J. S. Fell, B. N. Martin, K. N. Houk, *J. Org. Chem.* **2017**, *82*, 1912.
- [35] P. A. Champagne, K. N. Houk, *J. Org. Chem.* **2017**, *82*, 10980.
- [36] S. Roy, K. C. Mondal, J. Meyer, B. Niepötter, C. Köhler, R. Herbst-Irmer, D. Stalke, B. Dittrich, D. M. Andrada, G. Frenking, H. W. Roessky, *Chem. Eur. J.* **2015**, *21*, 9312.
- [37] A. Couce-Rios, A. Lledós, I. Fernández, G. Ujaque, *ACS Catal.* **2019**, *9*, 848.
- [38] A. P. Bento, F. M. Bickelhaupt, *J. Org. Chem.* **2008**, *73*, 7290.
- [39] B. Galabov, G. Koleva, H. F. Schaefer, III., W. D. Allen, *Chem. Eur. J.* **2018**, *24*, 11637.
- [40] J. M. Gonzales, C. Pak, R. S. Cox, W. D. Allen, H. F. Schaefer, III., A. G. Császár, G. Tarczay, *Chem. Eur. J.* **2003**, *9*, 2173.
- [41] W. J. van Zeist, C. Fonseca Guerra, F. M. Bickelhaupt, *J. Comput. Chem.* **2008**, *29*, 312.
- [42] ADF2017, SCM, Theoretical Chemistry, Vrije Universiteit, Amsterdam, The Netherlands, <http://www.scm.com>. E. J. Baerends, T. Ziegler, A. J. Atkins, J. Autschbach, O. Bascggio, D. Bashford, A. Bérces, F. M. Bickelhaupt, C. Bo, P. M. Boerrigter, L. Cavallo, C. Daul, D. P. Chong, D. V. Chulhai, L. Deng, R. M. Dickson, J. M. Dieterich, D. E. Ellis, M. van Faassen, L. Fan, T. H. Fischer, C. Fonseca Guerra, M. Franchini, A. Ghysels, A. Giammona, S. J. A. van Gisbergen, A. Goetz, A. W. Götz, J. A. Groeneveld, O. V. Gritsenko, M. Grüning, S. Gusarov, F. E. Harris, P. van den Hoek, Z. Hu, C. R. Jacob, H. Jacobsen, L. Jensen, L. Joubert, J. W. Kaminski, G. van Kessel, C. König, F. Kootstra, A. Kovalenko, M. V. Krykunov, E. van Lenthe, D. A. McCormack, A. Michalak, M. Mitoraj, S. M. Morton, J. Neugebauer, V. P. Nicu, L. Noodleman, V. P. Osinga, S. Patchkovskii, M. Pavanello, C. A. Peebles, P. H. T. Philipsen, D. Post, C. C. Pye, H. Ramanantoanina, P. Ramos, W. Ravenek, J. I. Rodríguez, P. Ros, R. Rüger, P. R. T. Schipper, D. Schlüns, H. van Schoot, G. Schreckenbach, J. S. Seldenthuis, M. Seth, J. G. Snijders, M. Solà, M. Stener, M. Swart, D. Swerhone, V. Tognetti, G. te Velde, P. Vernooijs, L. Versluis, L. Visscher, O. Visser, F. Wang, T. A. Wesolowski, E. M. van Wezenbeek, G. Wiesenecker, S. K. Wolff, T. K. Woo, A. L. Yakovlev.
- [43] C. Fonseca Guerra, J. Snijders, G. te Velde, E. J. Baerends, *Theor. Chem. Acc.* **1998**, *99*, 391.
- [44] G. te Velde, F. M. Bickelhaupt, E. J. Baerends, C. Fonseca Guerra, S. J. A. van Gisbergen, J. G. Snijders, T. Ziegler, *J. Comput. Chem.* **2001**, *22*, 931.
- [45] M. Handzlik, PLAMS (Python Library for Automating Molecular Simulations). Available at <https://www.scm.com/doc/plams/>. (Accessed May 24, 2019).
- [46] F. Zapata, L. Ridder, J. Hidding, QMflows. Available at <https://doi.org/10.5281/zenodo.1045523>. (Accessed May 24, 2019).
- [47] M. J. Frisch, G. W. Trucks, H. B. Schlegel, G. E. Scuseria, M. A. Robb, J. R. Cheeseman, G. Scalmani, V. Barone, B. Mennucci, G. A. Petersson, H. Nakatsuji, M. Caricato, X. Li, H. P. Hratchian, A. F. Izmaylov, J. Bloino, G. Zheng, J. L. Sonnenberg, M. Hada, M. Ehara, K. Toyota, R. Fukuda, J. Hasegawa, M. Ishida, T. Nakajima, Y. Honda, O. Kitao, H. Nakai, T. Vreven, J. A. Montgomery, Jr., J. E. Peralta, F. Ogliaro, M. Bearpark, J. J. Heyd, E. Brothers, K. N. Kudin, V. N. Staroverov, R. Kobayashi, J. Normand, K. Raghavachari, A. Rendell, J. C. Burant, S. S. Iyengar, J. Tomasi, M. Cossi, N. Rega, J. M. Millam, M. Klene, J. E. Knox, J. B. Cross, V. Bakken, C. Adamo, J. Jaramillo, R. Gomperts, R. E. Stratmann, O. Yazyev, A. J. Austin, R. Cammi, C. Pomelli, J. W. Ochterski, R. L. Martin, K. Morokuma, V. G. Zakrzewski, G. A. Voth, P. Salvador, J. J. Dannenberg, S. Dapprich, A. D. Daniels, Ö. Farkas, J. B. Foresman, J. V. Ortiz, J. Cioslowski, D. J. Fox, Gaussian 09, Revision E.01, Gaussian, Inc., Wallingford, CT, **2009**.
- [48] F. Neese, *WIREs Comput. Mol. Sci.* **2012**, *2*, 73.
- [49] F. Furche, R. Ahlrichs, C. Hättig, W. Klopper, M. Sierka, F. Weigend, *WIREs Comput. Mol. Sci.* **2014**, *4*, 91.
- [50] Bokeh, Available at <https://bokeh.pydata.org/en/latest/>. (Accessed May 24, 2019).
- [51] K. Fukui, *Acc. Chem. Res.* **1981**, *14*, 363.
- [52] L. Deng, T. Ziegler, L. Fan, *J. Chem. Phys.* **1993**, *99*, 3823.
- [53] L. Zhao, M. von Hopffgarten, D. M. Andrada, G. Frenking, *WIREs Comput. Mol. Sci.* **2018**, *8*, e1345.
- [54] T. Ziegler, A. Rauk, *Inorg. Chem.* **1979**, *18*, 1558.
- [55] T. Ziegler, A. Rauk, *Inorg. Chem.* **1979**, *18*, 1755.
- [56] A. Diefenbach, F. M. Bickelhaupt, *J. Chem. Phys.* **2001**, *115*, 4030.
- [57] W. J. van Zeist, A. H. Koers, L. P. Wolters, F. M. Bickelhaupt, *J. Chem. Theory Comput.* **2008**, *4*, 920.
- [58] J. Z. A. Laloo, N. Savoo, N. Laloo, L. Rhyman, P. Ramasami, *J. Comput. Chem.* **2019**, *40*, 619.
- [59] M. W. Schmidt, K. K. Baldrige, J. A. Boatz, S. T. Elbert, M. S. Gordon, J. H. Jensen, S. Koseki, N. Matsunaga, K. A. Nguyen, S. Su, T. L. Windus, M. Dupuis, J. A. Montgomery, Jr., *J. Comput. Chem.* **1993**, *14*, 1347.
- [60] M. S. Gordon, M. W. Schmidt, In Theory and Applications of Computational Chemistry: The First Forty Years; C. E. Dykstra, G. Frenking, K. S. Kim, G. E. Scuseria, Eds., Elsevier, Amsterdam, Netherlands, **2005**, p. 1167.
- [61] GAMESS (The General Atomic and Molecular Electronic Structure System). Available at <http://www.msg.ameslab.gov/games/index.html>. (Accessed May 24, 2019).
- [62] D. Svatunek, K. N. Houk, *ChemRxiv* **2019**, DOI: <https://doi.org/10.26434/chemrxiv.7789940.v1>.
- [63] M. Valiev, E. J. Bylaska, N. Govind, K. Kowalski, T. P. Straatsma, H. J. J. van Dam, D. Wang, J. Nieplocha, E. Apra, T. L. Windus, W. A. de Jong, *Comput. Phys. Commun.* **2010**, *181*, 1477.
- [64] Y. Shao, Z. Gan, E. Epifanovsky, A. T. B. Gilbert, M. Wormit, J. Kussmann, A. W. Lange, A. Behn, J. Deng, X. Feng, D. Ghosh, M. Goldey, P. R. Horn, L. D. Jacobson, I. Kaliman, R. Z. Khaliullin, T. Kuš, A. Landau, J. Liu, E. I. Proynov, T. R. Furlani, B. R. Brooks, D. G. Truhlar, T. Van Voorhis, A. I. Krylov, P. M. W. Gill, M. Head-Gordon, *Mol. Phys.* **2015**, *113*, 184.
- [65] J. F. Hartwig, Organotransition Metal Chemistry-From Bonding to Catalysis, University Science Books, Sausalito, CA, **2010**.
- [66] M. B. Smith, March's Advanced Organic Chemistry: Reactions, Mechanisms, Structure, Wiley, New York, **2013**.
- [67] E. I. Negishi, *Angew. Chem. Int. Ed.* **2011**, *50*, 6738.
- [68] A. Suzuki, *Angew. Chem. Int. Ed.* **2011**, *50*, 6722.
- [69] G. T. de Jong, F. M. Bickelhaupt, *ChemPhysChem* **2007**, *8*, 1170.
- [70] R. Huisgen, *J. Org. Chem.* **1976**, *41*, 403.
- [71] J. P. Collman, N. K. Devaraj, C. E. D. Chidsey, *Langmuir* **2004**, *20*, 1051.
- [72] A. E. Speers, G. C. Adam, B. F. Cravatt, *J. Am. Chem. Soc.* **2003**, *125*, 4686.
- [73] P. L. Golas, N. V. Tsarevsky, B. S. Sumerlin, K. Matyjaszewski, *Macromolecules* **2006**, *39*, 6451.
- [74] T. S. Seo, X. Bai, D. H. Kim, Q. Meng, S. Shi, H. Ruparel, Z. Li, N. J. Turro, J. Ju, *Proc. Natl. Acad. Sci. U. S. A.* **2005**, *102*, 5926.
- [75] J. C. Jewett, C. R. Bertozzi, *Chem. Soc. Rev.* **2010**, *39*, 1272.
- [76] X. Ning, J. Guo, M. A. Wolfert, G. J. Boons, *Angew. Chem. Int. Ed.* **2008**, *47*, 2253.
- [77] F. Schoenebeck, D. H. Ess, G. O. Jones, K. N. Houk, *J. Am. Chem. Soc.* **2009**, *131*, 8121.
- [78] M. Breugst, R. Huisgen, H. U. Reissig, *Eur. J. Org. Chem.* **2018**, 2477.
- [79] I. Fleming, J. P. Michael, L. E. Overman, G. F. Taylor, *Tetrahedron Lett.* **1978**, *19*, 1313.
- [80] T. A. Hamlin, M. Swart, F. M. Bickelhaupt, *ChemPhysChem* **2018**, *19*, 1315.
- [81] L. Deng, V. Branchadell, T. Ziegler, *J. Am. Chem. Soc.* **1994**, *116*, 10645.
- [82] F. M. Bickelhaupt, H. Hermann, G. Boche, *Angew. Chem. Int. Ed.* **2006**, *45*, 823.
- [83] R. Luschtinetz, J. Frenzel, T. Milek, G. Seifert, *J. Phys. Chem.* **2009**, *113*, 5730.
- [84] S. Grimme, C. Bannwarth, P. Shushkov, *J. Chem. Theory Comput.* **2017**, *13*, 1989.
- [85] X. Sun, T. M. Soini, J. Poater, T. A. Hamlin, F. M. Bickelhaupt. PyFrag 2019 Source. <https://github.com/sunxb05/PyFrag>. (Accessed May 24, 2019).
- [86] X. Sun, T. M. Soini, J. Poater, T. A. Hamlin, F. M. Bickelhaupt. PyFrag 2019 tutorial. <https://pyfragdocument.readthedocs.io/en/latest/includeme.html>. (Accessed May 24, 2019).

Received: 20 April 2019

Revised: 16 May 2019

Accepted: 16 May 2019

Published online on 4 June 2019