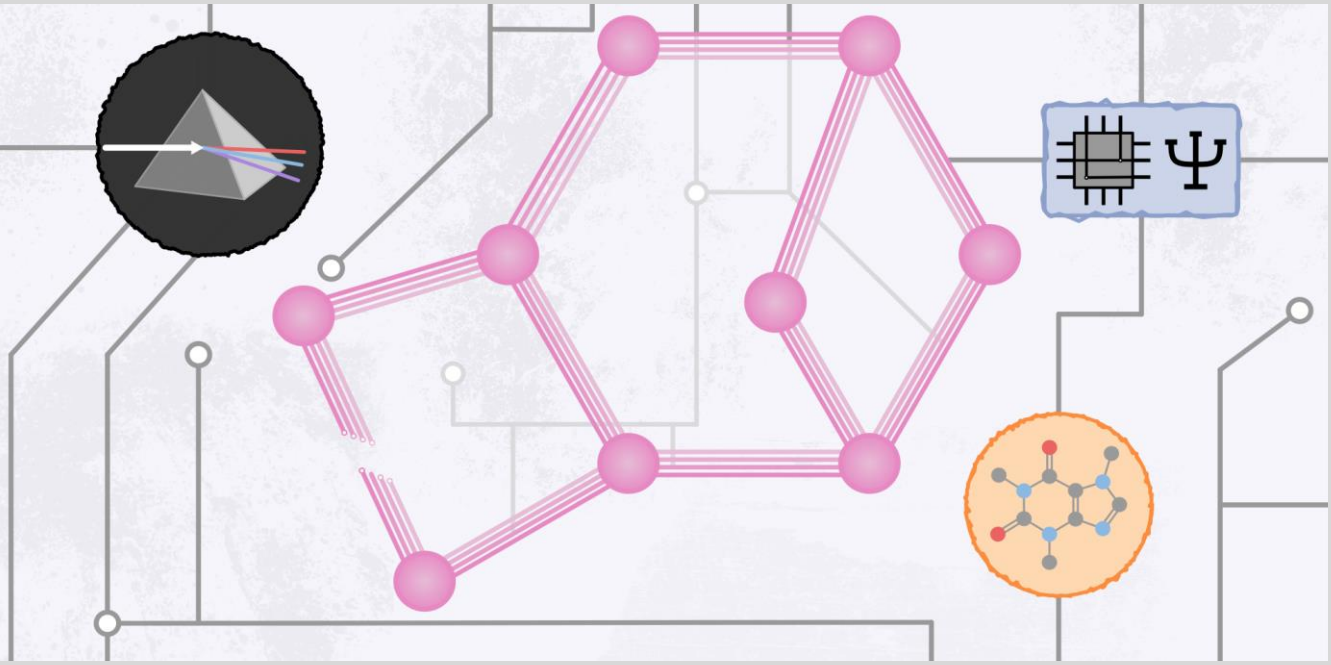
UNWEAVING COMPLEX REACTIVITY: GRAPH-BASED TOOLS TO HANDLE CHEMICAL REACTION NETWORKS

Diego Garay Ruiz

UNIVERSITAT ROVIRA i VIRGILI

Unweaving complex reactivity: graph-based tools to handle chemical reaction networks.

Diego Garay Ruiz



DOCTORAL THESIS
2023

Diego Garay Ruiz

Unweaving Complex Reactivity: Graph-Based Tools to Handle Chemical Reaction Networks

Doctoral Thesis
Supervised by Prof. Carles Bo Jané

Tarragona
2023

UNIVERSITAT
ROVIRA i VIRGILI

Dr. Carles Bo i Jané, investigador de l'ICIQ i professor de la Universitat Rovira i Virgili, FAIG CONSTAR que aquest treball titulat "Unweaving Complex Reactivity: Graph-Based Tools to Handle Chemical Reaction Networks", que presenta Diego Garay Ruiz per a l'obtenció del títol de Doctor, ha estat realitzat sota la meva direcció a l'Institut Català d'Investigació Química.

Signat per Carles Bo Jané - DNI 39671384H (TCAT)
13 Dec 2022 18:22:33 CET
Certificat emès per: EC-SectorPublic
Número de serie:
83773201961129976384666817767031334572
Fundació Institut Català d'Investigació Química (ICIQ)

El director de la tesi doctoral Carles Bo i Jané,
Tarragona, 13 de desembre 2022

Acknowledgments

From the many things that have to be written along a doctorate, surely the acknowledgments of the Thesis are among the most difficult (if not the most).

I shall begin by thanking Carles, first for giving me the chance to come to ICIQ to carry out this PhD that now comes to its end, but especially for being a really good supervisor. For all the support, for letting me try so many different things, and for believing that every and each of these things could be useful and valuable. Without this, the whole journey would have been much less interesting (and I would have learned much less).

Then, of course, I also have to thank all the labmates that have been around through these years and made the experience quite far from lonely, even with a year and a half of pandemic telework happening in the middle. Thanks also to all the people I have collaborated with in the myriad of projects I have attempted through the PhD, because at the end it is always about teamwork. And very special thanks to “the gnus”, for being, by far, the best support that I could have here.

Now going farther from Tarragona, I must also thank my parents, for always supporting me even if it meant that I could not be home more than three or four times a year, and for listening whenever I had something to rant about. Same thing can be said for the friends from Aranda and from Valladolid, going between our scarce real-life encounters by an uncountable number of voice notes.

Last but not least, I would also like to thank Carmen, Fermín and Ignacio for paving the scientific pathway that eventually has took me here, showing me what computational chemistry was and how many cool things could be done through it.

Funding

This work has been funded by the Institute of Chemical Research of Catalonia (ICIQ) and by a FI grant from the Secretaria d'Universitats i Recerca de la Generalitat de Catalunya and the European Social Fund.

ICIQ
Institut
Català
d'Investigació
Química

UNIVERSITAT
ROVIRA i VIRGILI

Generalitat de Catalunya
Departament de Recerca
i Universitats

Unió Europea
Fons social europeu
L'FSE inverteix en el teu futur

GOBIERNO
DE ESPAÑA

MINISTERIO
DE CIENCIA
E INNOVACIÓN

List of publications

1. **Effect of an Al(III) Complex on the Regio- and Stereoisomeric Formation of Bicyclic Organic Carbonates**, Cristina Maquilón, Bart Limburg, Victor Laserna, Diego Garay-Ruiz, Joan González-Fabra, Carles Bo, Marta Martínez Belmonte, Eduardo C. Escudero-Adán and Arjan W. Kleij, *Organometallics* (2020), 39, 9, 1642 – 1651.

   DOI 10.1021/acs.organomet.9b00773

2. **Revisiting Catalytic Cycles: A Broader View through the Energy Span Model**, Diego Garay-Ruiz and Carles Bo, *ACS Catalysis* (2020), 10, 21, 12627 – 12635.

   DOI 10.1021/acscatal.0c02332

3. **Rationalizing the Mechanism of Peroxyformate Decomposition: Computational Insights to Understand Solvent Influence**, Diego Garay-Ruiz and Carles Bo, *Chemistry – A European Journal* (2021), 27, 45, 11618 – 11626.

   DOI 10.1002/chem.202100755

4. **New Tools for Taming Complex Reaction Networks: The Unimolecular Decomposition of Indole Revisited**, Diego Garay-Ruiz, Moisés Álvarez-Moreno, Carles Bo and Emilio Martínez-Núñez, *ACS Physical Chemistry Au* (2022), 2, 3, 225-236.

   DOI 10.1021/acsphyschemau.1c00051

5. **Chemical Reaction Network Knowledge Graphs: the OntoRXN Ontology**, Diego Garay-Ruiz and Carles Bo, *Journal of Cheminformatics* (2022), 29.

   DOI 10.1186/s13321-022-00610-x

Contents

List of Abbreviations

API Application Programming Interface

BBFS Bond Breakage Formation Search

BFS Broadth-First Search

CML Chemical Markup Language

CRN Chemical Reaction Network

DFS Depth-First Search

DFT Density Functional Theory

DOI Digital Object Identifier

ESM Energy Span Model

HF Hartree-Fock

HTML Hypertext Markup Language

iCOM Interstellar Complex Organic Molecule

InChI IUPAC International Chemical Identifier

ISM Interstellar Medium

KG Knowledge Graph

KMC Kinetic Monte Carlo

LH Langmuir-Hinshelwood

MD Molecular Dynamics

MESA Modified Energy Span Analysis

ML Machine Learning

OBO Open Biomedical Ontologies

OWL Web Ontology Language

PES Potential Energy Surface

RDF Resource Description Framework

REST API Representational State Transfer Application Programming Interface

RRKM Rice-Ramsperger-Kassel-Marcus Theory

RXNet Reaction Network

SPARQL SPARQL Protocol and RDF Query Language

SQL Structured Query Language

TOF Turnover Frequency

TON Turnover Number

TS Transition State

TST Transition State Theory

URI Uniform Resource Identifier

URL Uniform Resource Locator

XML eXtended Markup Language

XSLT eXtended Stylesheet Language Transformation

Abstract

The characterization of reaction mechanisms is an essential asset in Chemistry, enabling a better understanding and thus a much more rational optimization of the underlying processes. In this context, *in silico* studies, providing profound insights at the atomistic and molecular levels, have become a key part of mechanistic elucidations. The breakthrough of computational chemistry has been fueled by the constant increase in computational power taking place along the last decades, as it has vastly increased its overall applicability. In this sense, we may consider not only increases in the size of the systems that can be studied or the accuracy of the available methods, but also in the level of detail of the resulting mechanistic description, characterizing larger numbers of possible intermediates and transformations. This is often attained through the use of automation techniques that simplify the exploration of the vast chemical space. However, enhancing the description of reaction mechanisms comes at the cost of increasing their complexity, and thus hindering their interpretation: it then becomes necessary to develop strategies that automate not only the collection of results, but also its further processing in order to really leverage these larger volumes of data. Throughout this Thesis and following this main principle, we have proposed, developed and tested a set of tools for the processing of this kind of complex mechanisms, represented as Chemical Reaction Networks (CRNs), to obtain better insights on reactive and catalytic processes. As a general note, all the tools introduced in the Thesis model the target CRNs as *graphs*, applying some of the methods of Graph Theory (e.g. path searches, isomorphisms...) under a chemical context.

First of all, **amk-tools** (*Chapter III*) provides a solution to filter and visualize the complex CRNs generated by automated reaction mechanism discovery programs, such as AutoMeKin. The combination of network trimming, isolating only the most chemically relevant sections of the CRN, and interactive visualization provides a clearer perspective on the highly complex networks arising from this kind of highly automated approaches, exemplifying the aforementioned relationship between detail and complexity. Then, moving to the topic of catalysis, **gTOFfee** (*Chapter IV*) aims to obtain a direct picture of the overall activity of complex catalytic cycles by applying the energy span model to CRNs, computing their turnover frequency (TOF) and effective energy span (δE_{eff}). These two magnitudes take the complete cycle into account at once, in order to capture the influence of all intermediates and transition states participating in the system. Finally, **OntoRXN** (*Chapter V*) is proposed as a data organization scheme to manage all the information related to the computational characterization of CRNs, putting together the interconnectivity of the different states of the system and the properties calculated by electronic structure methods for every structure participating in the network. In this way, OntoRXN allows to apply the principles of Semantic Data to this kind of systems, producing scalable, expandible and versatile *knowledge graphs* as a standardized format to share and utilize reaction networks.

Chapter I

Introduction: chemical reactivity, catalysis and computational chemistry

1.1 Chemical reactivity

Reactivity has been at the heart of Chemistry since the earliest beginnings of this discipline. Indeed, we may somehow trace it back even *before* chemical science itself, as the notion of "transmutation" coined by alchemists has undeniable resemblances to the current notion of a chemical reaction: the transformation of a given substance to a totally different one. Either thinking under the old framework of alchemy or the modern framework of chemistry, large parts of human history have been undoubtedly marked by the development of new transformations between substances to create novel products. To be more precise and following the definition from the International Union for Pure and Applied Chemistry (IUPAC) [1], a chemical reaction is simply "a process resulting in the interconversion of chemical species", a broad term that engulfs the whole spectrum of processes from the simplest elementary proton shifts happening between an acidic and a basic substrate, to the highly complex cascade transformations mediated by enzymes that enable biochemistry, metabolism and life. Thus, studying how chemical reactions take place is a centerpiece for any advancement happening in not only pure chemistry, but also in pharmacy or biology, among several other examples.

The main difference between chemistry, as a science, and its pre-scientific background is the *rationalization* of the observations that are made. Apart from the mere discovery of new substances, transformations or phenomena, major efforts are devoted to hypothetize the reasons behind the new findings and to contextualize those in the framework of previous theories. Albeit setting an exact date of separation between the labels of alchemy and chemistry is not feasible, as the development of the latter from the roots of the former was gradual, we may consider the 17th century and the enunciation of the scientific method by Francis Bacon as a stepping stone. From there on, a bunch of essential discoveries and noteworthy scientists proceed, such as the explanation of the behavior of gases (Boyle), the isolation and discovery of hydrogen (Cavendish) or oxygen

(Scheele, Priestley) and, especially, the thorough development of chemical nomenclature and compilation of the current knowledge made by Antoine de Lavoisier. The work that Lavoisier carried out at the end of the 18th century provided a solid foundation that enabled the further development of chemistry to the science we know nowadays. Then, in the early 19th century Dalton eventually proposed the first atomic theory, proposing matter to be formed by microscopic, undivisible entities (atoms) of different nature (chemical elements). This, followed by the introduction of modern chemical symbols and the extensive study of atomic weights by Berzelius, conformed another crucial part of the scaffold in which current chemistry builds on.

Speaking about the rationalization of chemistry in the framework of atomic theory, perhaps the clearest example is the development of the Periodic Table by Mendeleev on the second half of the 19th century. This profound systematization of the different chemical elements and its properties was able not to only classify known information, but also to accurately predict the properties of undiscovered elements in a pristine example of a successful scientific model, which is still on use as of today. Along the same period of time, the studies from Boltzmann and Gibbs coined terms as important in physical chemistry as entropy, enthalpy or free energy, creating and driving forward the concept of statistical mechanics as another bridge between the microscopic and macroscopic descriptions of matter. Later on, in the first half of the 20th century, the whole physical science was revolutionized by the introduction of quantum mechanics (de Broglie, Heisenberg, Schrödinger, Pauli), deeply impacting chemistry with a solid theoretical and mathematical framework. Once this theoretical basis was stated, allowing to properly enounce fundamental chemistry in mathematical terms, the major point left to address was to actually *solve* the intricate equations arising from quantum mechanics: the main challenge for computational chemistry, which will be introduced in Section 1.3.

Whilst this brief historical overview refers to chemistry in general, it is also directly linked with the understanding of reactive processes. Lavoisier's quantitative studies introduced the central notion of reaction stoichiometry, which in combination with Dalton's atomic theory and Berzelius' chemical symbols allowed, for the first time, to define something close to our current idea of a balanced chemical reaction. Then, passing to a more physico-chemical lens, the thermodynamic potentials from Boltzmann or Gibbs allow to justify the feasibility or spontaneity of reactions. This, in connection with the quantum-based approach of theoretical chemistry, provides the framework for the computational-based description of reaction mechanisms that is central to this Thesis.

1.2 Catalysis and its role on society

Within the realm of chemical reactivity, a field of particular interest and importance is that of *catalysis*. According to the IUPAC [1], catalysis is defined as “the action of a catalyst”, which is “a substance that increases the rate of a reaction without modifying the overall standard Gibbs free energy change”. A catalyst is then, at the same time, a reactant and a product of the reaction, with no net catalyst consumption happening along the process. Catalysts do not only accelerate slow reactions (by lowering the activation barrier of the process, as depicted in Figure 1.1), but also enable reactions that would not take place at all in other conditions or modify the kind of products generated from a given substrate, thus providing a powerful degree of control over how a chemical process occurs.

The extent of control and optimization provided by catalytic processes has made them a completely essential asset for the chemical industry. Currently, between a 80 and a 90 % of all chemicals are produced through catalytic processes [2, 3], making their importance clear. A paradigmatic example of how influential catalysis can be to humankind is the development of the Haber-Bosch process in the 1930s, which enabled the fixation of nitrogen from air in the presence of iron catalysts (Equation 1.1) to produce trivalent nitrogen in the form of ammonia, which can be used to produce fertilizers. The availability of these fertilizers permitted a massive growth in the production of crops that was key in the increase of the global population

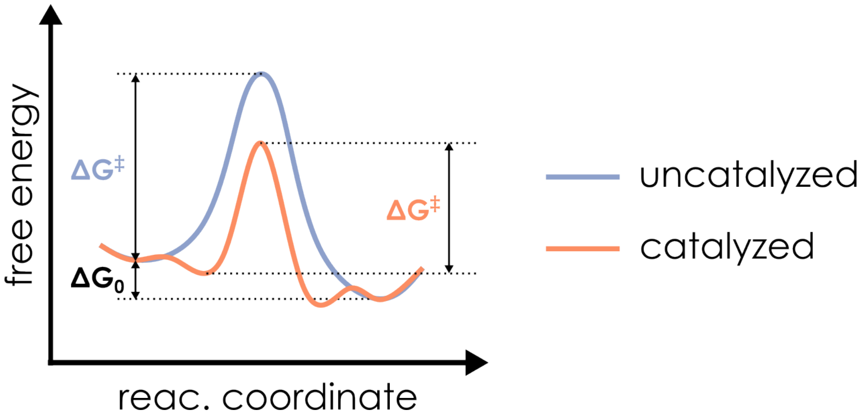


Figure 1.1: Example of the free energy profile for a chemical reaction $A + B \rightarrow C$, with (orange) and without (blue) a catalyst.

along the 20th century.

$$N_2 + 3\ H_2 \xrightarrow[500^{\circ}C]{Fe} 2\ NH_3 \tag{1.1}$$

The most common subdivision of catalysts refers to the phase in which they reside in the reactive mixture, distinguishing between homogeneous and heterogeneous catalysis. Regarding industrial applications, heterogeneous catalysts are the most widely used (around an 80% of the total) due to their easier recovery and increased stability, as consequences of the phase separation. However, homogeneous catalysis shall not be neglected, as it provides a finer degree of control over the products that are obtained and permits a more complete mechanistic understanding of the underlying processes, eventually facilitating the optimization of the catalytic species to improve the overall yields or selectivities. Hitherto, both catalytic paradigms still coexist to date and bring their own advantages (and limitations) to the table. Regardless of the phase of operation, most of the catalysts that are currently in use, particularly for industrial applications, involve metal atoms (either as part of surfaces or forming organometallic complexes), although metal-free approaches such as organocatalysis are becoming increasingly relevant.

The aforementioned Haber-Bosch process and its consequences also exemplify one of the major challenges that science, chemistry and catalysis face nowadays: the need to develop greener, environment-friendly processes in times of an undeniable anthropogenic climate change [4–6]. Producing ammonia through the Haber-Bosch method, even after almost a hundred years of development and improvement, is a quite energy-intensive process which supposes the consumption of large quantities of fossil fuels (gas, oil, coal), corresponding on its own to an 1-2% of the total emissions of carbon dioxide to the atmosphere [7–9]. This carbon footprint could be reduced by using renewable energy sources instead of fossil fuels or by optimizing the ammonia-producing system (Eq. 1.1), but also by having cleaner and more efficient routes to generate hydrogen. This species is usually generated from methane in another energetically costly and wasteful process, which adds to the overall footprint of the Haber-Bosch approach. Precisely, the development of efficient catalysts for hydrogen production (usually via *water splitting* $H_2O \rightarrow H_2 + 1/2\ O_2$) is an area of the uttermost interest in current catalysis, not only as a reagent for industrial and fine chemicals, but as a sustainable, clean energy source [10, 11] to replace traditional fuels.

Along with the production of "green" hydrogen, we may highlight several other major challenges for catalysis that are also framed in the context of environmentally conscious chemistry: for instance, the search for cleaner and more efficient processes to produce fine chemicals or the reutilization of reaction subproducts to reduce the amount of released waste. Regarding the first aspect, we encounter a clear example on the Nobel Prize of Chemistry awarded in 2021 [12] to List and McMillan for the development of asymmetric organocatalysts, which enable the enantioselective synthesis of complex, chiral molecules without having metal atoms in the catalytic species. In this way, it becomes possible to avoid the scarcity or toxicity issues that can be related with metal-based catalysts while also obtaining enantiopure products, of sheer importance in pharmacy and medicine. As for the latter challenge, a paradigmatic example is the fixation of carbon dioxide, an ubiquitous subproduct for any process involving the combustion of organic molecules which is one of the major contributors to greenhouse effect and global warming. The development of catalytic strategies to employ CO_2 as a synthetic building block [13–15] allows not only to reduce overall carbon emissions but also to have access to a versatile C1 synthon that can open the door to powerful and controlled synthetic routes to industrially relevant products.

All in all, catalysis arises as one of the most powerful tools that chemistry provides to tackle the growing issue of climate change and bring industry and society forward to a cleaner and livable environment. To be able to do this, and in line with Section 1.1, it is essential to have a better *understanding* on how catalytic processes work, to find the aspects that can be tweaked and become able to rationally design novel functioning, well-performing catalysts to attain all of these ambitious goals. Nowadays, a proper rational design implies an interplay between carefully designed experiments, precise characterization and the assistance of computational chemistry to unveil the underlying catalytic mechanisms at the molecular level.

1.3 Computational chemistry to understand reactivity and catalysis

The label “computational chemistry”, in the most general sense, refers to any chemical problem that is tackled and solved through the use of computers. This includes very distinct tasks, such as chemical database design, computer-aided synthetic route generation, quantitative structure-activity relationships (QSAR) to predict molecular properties, or simulations of molecules and materials aiming to model their behavior at the microscopic level. Nowadays, the more data-related approaches (those that deal with *chemical information*, such as the three first elements in the previous enumeration) are often designated under the term “cheminformatics” [16], with “computational chemistry” being more used to refer to the more simulation-based approaches. Nonetheless, the actual distinction is fuzzy, especially considering how data-based Machine Learning methods are gaining importance in the field not only to predict specific properties (similar to the aforementioned QSAR methods) but also as a driver for atomistic simulations [17–19].

Among the very diverse methods of computational chemistry, we will focus on *electronic structure* methods, approximating the solution of the Schrodinger equation $\hat{H}\psi = E\psi$ for the multi-electron set of a given chemical system. Through these approximations, it becomes possible to predict not only the raw *electronic energy* arising from the equation itself, but also the optimal geometry (or a set of them), vibrational frequencies, thermochemical properties, etc. For high-accuracy results, there are two main paradigms to tackle the electronic structure problem: *ab initio* methods, centered on the wavefunction ψ of the system, and Density Functional Theory (DFT) methods, which relate the energy to the electron density ρ.

No matter the specific kind of electronic structure elucidation that is chosen, there is an essential approximation at the core to deal with multielectronic systems: the Born - Oppenheimer approximation, which decouples electronic motion from nuclear motion. In this way, the electronic wavefunction ψ_e is assumed to be independent of the momenta of the nuclei, just depending on their fixed positions. Reversing the point of view, this implies assuming that nuclei move through a *potential energy surface* (PES), introducing an essential concept to discuss chemical reactivity from the quantum-mechanical perspective (Figure 1.2). Slow or unfeasible transformations correspond to nuclear displacements through high-energy regions of the PES that are difficult to access, while faster reactions are associated with paths that are easier to traverse and allow the nuclei to reorganize. The minima of the PES correspond to the most stable atomic arrangements and therefore to actual chemical entities, transforming the problem of finding stable entities (e.g. molecules) from quantum calculations to an *optimization* problem over the PES. However, this apparent simplicity is hindered by the large dimensionality of the surface in question, which

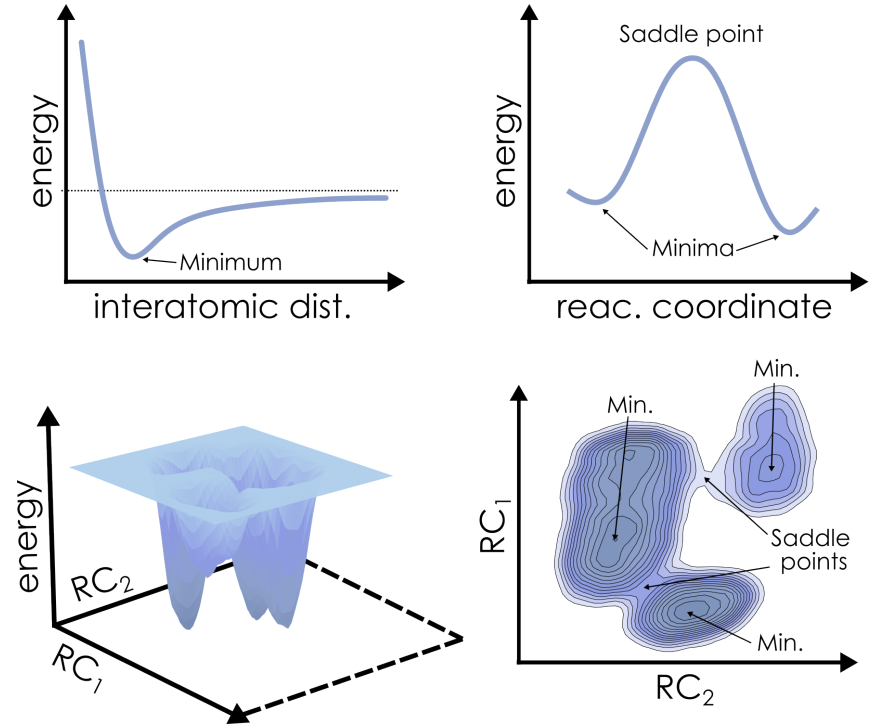


Figure 1.2: Examples of 2D (above) and 3D (below) potential energy surfaces. Above left, typical curve for the dissociation of a diatomic molecule. Above right, model for an unimolecular reaction step going through a transition state (saddle point). Below, 3D surface view (left) and contour map (right), with darker colors corresponding to lower energies, for a system with three minima and two transition states.

depends on a total of **3N - 6** coordinates[1]: three positional coordinates per each of the **N** atoms in the system, minus three degrees of freedom for the overall translation of the molecule as a whole across the space, minus three more for rotation. Representing and interpreting PES directly is, generally, not a feasible approach, as in all cases but the diatomic one ($N = 2; d = 3 \cdot 2 - 5 = 1$, upper left corner of Fig. 1.2) the resulting surfaces will always be hypersurfaces. For instance, triatomic molecules $N = 3; d = 3, 4$ (for non-linear or linear molecules) already produce 4D

[1] In the case of linear molecules, there are only two rotational degrees of freedom, and the PES involves only **3N - 5** coordinates

and 5D entities that cannot be depicted. Thus, it is more common to work with projections of reduced dimensionality across a set of one or two reaction coordinates to get the kind of 2D and 3D plots and maps shown in Figure 1.2. In fact, characterizing complete surfaces would be an incredibly time-consuming task, except for very small systems. Thus, computational chemistry focuses instead in solving the aforementioned optimization problem to locate critical points on the PES. Among these critical points, apart from the minima (stable structures), the first-order saddle points, which are minima in **d - 1** directions and maxima in the remaining one, are also deeply important. These saddle points, characterized by having a single negative second derivative (and thus, a single imaginary vibrational frequency) correspond to the *transition state* (TS) that governs the transformation happening between two minima, marking the lowest-energy pathway from one to the other. The concept of TSs is the core of Transition State Theory (TST), introduced by Eyring [20], where statistical thermodynamics are applied to the activated complex (TS) just like for intermediates, only that treating the imaginary vibrational normal mode of imaginary frequency as a translational degree of freedom. This allows to define the Eyring equation (Equation 1.2), where the rate constant of a given elementary step depends on the difference in free energy between the TS and the previous (or following, for the reverse rate) intermediate $\Delta G^{\ddagger}$.

$$k_{reac} = \frac{\kappa k_B T}{h} e^{\Delta G^{\ddagger}/RT} \tag{1.2}$$

Here, apart from the aforementioned activation free energy $\Delta G^{\ddagger}$, the temperature and the constants h, $\mathrm{k_B}$ and R, there is the coefficient κ that accounts for the transmission across the TS. The transmission coefficient is usually assumed as unity, meaning that no recrossing occurs at the barrier, while lower values would introduce some degree of deviation from the ideal theoretical limit of TST. As a consequence of this framework, characterizing these first-order saddle points in the PES gives access to the *kinetics* of the system, complementing the *thermodynamic* information arising from the study of the minima.

1.3.1 Overview of methods

As stated before, there are two main families of quantitative methods in computational chemistry: *ab initio* and DFT. We will only introduce a very basic overview of their foundations, referring to specific textbooks for more detailed, mathematically-ridden descriptions [21, 22]. First, *ab initio* methods employ only first principles to solve the Schrödinger equation under the Born-Oppenheimer approximation, not introducing any kind of empirical information. The "parent" method of this collection is the Hartree-Fock approach, which is based on a set of self-consistent equations (SCF or self-consistent field) that can be solved iteratively to obtain a suitable set of *molecular orbitals* (MOs), starting from a set of guess functions χ_i. The final set of MOs, once consistency between two consecutive iterations is reached, can be used to build an electronic wavefunction for the ground state, applying the eigenvalue equation $H_{el}\psi_{el} = E_{el}\psi_{el}$ to obtain the electronic energy.

The set of guess functions χ_i employed to approximate the molecular orbitals is named a *basis set*, and its choice comes to be an important part of the setup of a given computation scheme. Most of the basis sets employed in electronic structure codes contain Gaussian Type Orbitals (GTOs), which are mathematically convenient for the calculation of interaction integrals along the computational procedure. Other proposals such as Slater Type Orbitals (STOs) do also exist (and are implemented in codes like ADF) providing a better description of atom-level behavior with a smaller number of functions, at the cost of a more contrived integration. In terms of basis set size, it is common to include additional functions on the set on top of the minimum number that would be required to define the electrons in the system. This allows to improve the description of the electron distribution across the molecule, given that it might be very different from the distribution in the isolated atomic orbitals.

From this base approach, plenty of post-HF methods have been proposed along the years to improve the accuracy of *ab initio* methods, aiming to overcome the main drawback of HF, which is the mean-field treatment of electron-electron interactions. For each electron in the system, only an average repulsion with the other **n - 1** electrons is taken into account, not including explicit correlation effects which might have important effects on the predicted energies. Among these post-HF methods, we may mention configuration interaction (CI), which explicitly includes electron excitations when building the wavefunction, the n-order perturbative treatments by Moller and Pleset (MPn) or the coupled cluster (CC) methods that introduce excitations through an exponential operator. Currently, the CCSD(T) method (coupled cluster including simple and double excitations and approximating the more expensive triple excitations) is currently regarded as the “gold standard” for accuracy in computational chemistry. It should be recalled that increased accuracy comes with the deeply important downside of an increased computational cost that scales rapidly with the size of the system, going from around N^4 in HF to N^7 for CCSD(T).

On the other hand, DFT methods are founded in the Hohenberg-Kohn theorem [23], enounced in 1964, which states that the energy of the ground state of any electronic system is related to its electron density $\rho(r)$ through an exact, but unknown, functional **F**: $E = F[\rho(r)]$. One year later, Kohn and Sham [24] developed a set of self-consistent equations to apply the DFT formalism to multielectronic systems in analogy with the Hartree-Fock self-consistent field (SCF). Conceptually, DFT supposes a major reduction in the number of variables to be treated, going from the $\mathbf{3N + N}$ (spatial and spin) coordinates required to explicitly describe electrons to the system-independent three variables of the electron density. However, working under the Kohn-Sham paradigm, orbitals are introduced back in the formalism to improve the accuracy of the method, and thus the scaling comes back to be size-dependent, reaching N^3.

The main caveat of DFT applications is that the exact, universal functional introduced in the theoretical definition of the method is not known, so in practice it becomes necessary to work with approximate functionals to get the energy from the electron density. A wide variety of such functionals have been proposed since the first descriptions of DFT, incorporating different aspects and modifications (Table 1.1) in order to improve their accuracy throughout the quest for the definitive universal functional which properly considers electron exchange and correlation.

Category	Add. variables	Functionals
LDA	$\rho(r)$	LDA
GGA	$\nabla\rho(r)$	PBE, BP86, PW91
mGGA	$\nabla^2\rho(r), \tau(r)$	TPSS, M06L
Hybrid	HF exchange	B3LYP, PBE0, M06-2X, ωB97XD
Double hybrid	MP exchange	B2PLYP, DSD-PBEP86

Table 1.1: Main DFT functional categories: entries lower in the table are higher in the "Jacob's Ladder" depiction proposed by Perdew [25] to set a hierarchy for functionals, and thus shall be closer to chemical accuracy (universal functional). For each category, the principal included feature and some examples of common functionals are provided.

Even taking into account this classification, the choice of a specific functional for a specific task is not obvious, and either GGA, mGGA, hybrids or double hybrids might be adequate depending on the situation. In general, when there is good-quality reference data (experimental or highly accurate *ab initio* results) a common practice in computational chemistry is *benchmarking* a given system with different functionals to ensure that the chosen treatment can properly model the target system by, for example, reproducing known experimental properties.

1.3.2 Modern computational chemistry: automation, Big Data, Machine Learning

Computing, in general, has been a discipline marked by a very rapid development, as acknowledged by Gordon Moore in 1965 [26] by stating how the number of transistors integrated on a single chip shall, approximately, double every year (Moore's Law). This claim was later revised in 1975 [27] to extend the doubling period to two years, but has since then remained true until very recently [28]. The exponential increase in the number of transistors translates directly to an exponential increase in the overall computing power, enabling computers to carry out calculations and simulations of larger and larger scales.

Whilst this situation is immediately applicable to computational chemistry, there are several ways in which increased computer power can be employed. Following from the previous discussion, it becomes possible to employ more accurate methods that would be deemed as too expensive under less performing hardware. Due to the direct connection of computational cost and system size, the treatment of larger-sized entities does also become feasible, allowing computational chemists to use more realistic models and therefore capture effects that might have been overlooked in absence of these resources. Another avenue which is related to this kind of fine-grained effects is the possibility of performing massive sets of computations characterizing, for example, substituent variations over a given reaction scheme or catalytic cycle to determine how they affect the overall process.

The use of the word “massive” in the previous paragraph already hints at one of the main issues that we can encounter in this context: the human-driven setup becomes the bottleneck of the whole pipeline, limiting the extent that a given study can reach. Consequently, a key concept for modern computational chemistry (and many other fields of computational science) is *automation*: instructing the machine to perform routinary, repetitive tasks that can be performed without human intervention. This does not only strongly speed up the target process, but also diminishes the possible sources of error that people may introduce when performing a given task. Of course, this is a very general concept, and common electronic structure codes do already automate very complex tasks such as the application of the Hartree-Fock or Kohn-Sham methods described before and their integration with algorithms for exploring the PES, calculating thermodynamic potentials, etc (Figure 1.3, above). These procedures can then be included in workflows that automate modifications over molecules or other chemical entities, the

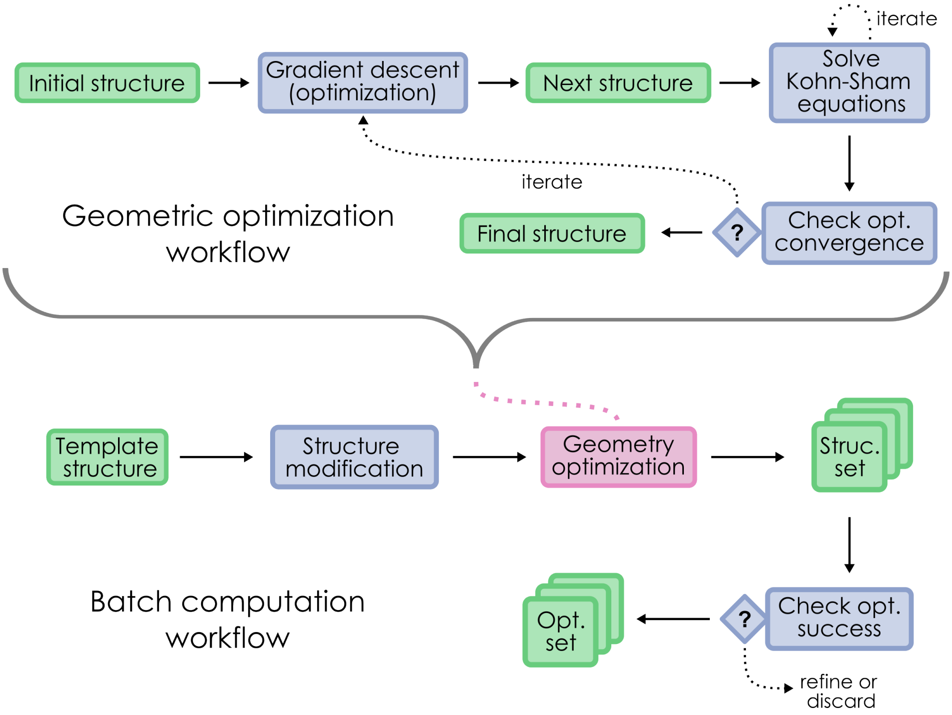


Figure 1.3: Schematic depiction of automation workflows in computational chemistry. Above, simplified procedure for a single geometry optimization task as carried out by a DFT electronic structure code. Below, example of substituent modification process generating a batch of structures starting from a template.

generation of the corresponding input files, and even the inspection of the correctness of the obtained results (Figure 1.3, below). While it is common to implement this kind of automation processes through tailor-made scripts, there are also workflow managers such as FireWorks [29] that simplify the setup and integration of complex procedures.

Nevertheless, it must be recalled that the main labor of a scientist (computational or not) is not to produce data, but to *interpret* and *analyze* that data to eventually extract some piece of novel knowledge from it. When the amount of raw information that is available increases, the analysis methods that would be used for smaller datasets may not be adequate anymore, making it necessary to add some kind of automation to the interpretation process too. For instance, we may consider a traditional use case for computational chemistry, such as the characterization of the free energy profile associated to a given chemical transformation. When the target of the study is a single energy profile, it can be tackled in a state-per-state basis, inspecting the individual geometric parameters, energies and other properties of interest along the transformation manually. However, if additional variations over that profile are considered to expand the scope of the study (modifying the computational method, the structure of reactants and products, catalytic species... and so on), a fully human-driven inspection becomes increasingly cumbersome, supposing a new bottleneck on the pipeline and eventually becoming unfeasible. Thus, as the degree of automation in data collection grows, it becomes necessary to also automate how the data is handled, building actual data manipulation pipelines. This situation is by no means exclusive to computational chemistry, as practically all areas of physical and social sciences are currently dealing with the need of managing exponentially volumes of information: the so-called Big Data [30, 31]. This term does most often come together with the idea of *Machine Learning* (ML), involving the development of methods to automatically detect patterns across large datasets through statistical analysis and infer knowledge from these patterns. Along the last few years, ML has quickly become a cornerstone in modern science [32], having impact in fields as diverse as economy, biology or medicine. Coming back to chemistry, we also encounter plenty of ML application examples for drug discovery [33], catalysis [34, 35], materials science [36] and many others. Nevertheless, despite the undeniable interest and growing impact of fully data-driven science, it still coexists with "traditional" theory-based modeling paradigms (e.g. *ab initio* or DFT methods in the case of computational chemistry), which, especially when fueled by modern automation protocols and computational power, provide irreplaceable insights on the behavior of physical and chemical systems.

1.4 Aims and objectives

The main goal of this Doctoral Thesis is to improve the interpretation, analysis and organization of the data gathered through computational chemistry methods, transforming the large amounts of information produced by increasingly automated pipelines into actual chemical knowledge. More specifically, we will be focusing on the study of reactivity and catalysis, defining complex *chemical reaction networks* to be processed by a set of new software tools in order to simplify the understanding of the likely complex underlying physicochemical processes. From this general goal, some more concrete objectives can be proposed.

- Identification and detailed characterization of chemical mechanisms of interest.
- Definition of a standard hierarchy and structure for the generation of chemical reaction networks from computational results.
- Integration of structural and energetic information for reaction network visualization.
- Calculation of descriptors related to the overall catalytic activity of complex networks.
- Propose a suitable organization scheme to integrate computational chemistry results and the structure of reaction networks in a machine-readable format.
- Development of automated workflows based on the previous goals.

First of all, in Chapter II we will do an overview of the theoretical background of this work, focusing on the definition of chemical reaction networks (CRNs) and its contextualization as part of a more general discussion on the representation of chemical reactivity. Later on, we will introduce some concepts of Graph Theory, one of the main tools employed along the Thesis to model and handle CRNs, as well as a brief description of Python, the language of choice for all the codes we developed. Finally, we will also comment on aspects more related to data organization, introducing the notion of Semantic Data and the ioChem-BD platform.

In Chapter III we will present **amk-tools**, a code for the interactive visualization of automatically discovered reaction networks integrated with the AutoMeKin package from Martínez-Núñez [37] for the generation of such mechanisms. Along the chapter, we will comment on the interest of this kind of visual approaches to make automated methods more accessible and appealing, detail the organization of the code and apply it to a specific target system: the unimolecular decomposition of indole.

Chapter IV introduces **gTOFfee**, a code for the application of the *energy span model* from Kozuch and Shaik [38] to any kind of catalytic cycle, leveraging the recent reformulation of the model in terms of graphs. Herein, we will discuss model implementation and the application of the Graph Theory concepts introduced in Chapter II to design a working code. To demonstrate the interest of our approach, we will apply gTOFfee to the cobalt-catalyzed hydroformylation of propene as a well-studied model system for homogeneous catalysis. As a final note, we will comment on the perspectives of employing this graph-based approach to improve the applicability of the energy span model to heterogeneous catalysis.

Through Chapter V we will delve on the application of Semantic Data to CRNs, reviewing the major developments on chemical ontologies before presenting our own take on the field: the **OntoRXN** ontology. We will then showcase the core structure of the ontology and how it may be used to describe the different entities and concepts required to properly define a reaction network from computational information. At the same time, we also present the set of tools (**ontorxn-tools**) that were built to simplify the direct application of OntoRXN to real datasets as stored in the ioChem-BD database. From there, we applied this approach to three reaction mechanisms, presenting a collection of example workflows utilizing OntoRXN-based knowledge graphs to carry out processes like the construction of complex reaction energy profiles or the automated preparation of microkinetic simulations.

Chapter II

Methodology & theoretical background: reaction networks, graph theory and data management

2.1 Concepts on reactivity

The treatment of chemical reactivity, whose importance has already been discussed along Chapter I, has given rise to multiple alternative representations through the years. This variety comes from the several facets of the study of reactive phenomena, ranging from general synthetic transformations (reagents, products and reaction conditions) to highly detailed reaction mechanisms interlinking plenty of elementary steps. Because of this multifaceted behavior, and also taking into account the widely different perspectives acquired from different subfields (e.g. syntheses, kinetic studies, computational studies...), these alternative representations end up as coexisting paradigms, having their own advantages and disadvantages and showing different degrees of transferrability across target subfields.

In our case, we are tackling the computational characterization of reaction mechanisms and catalytic cycles, prompting for a balance between thoroughly exposing the data that has been computed (mainly, energies) and linking these results with reference experimental studies to contextualize and validate the resulting models. In this context, we might think of three main representations, involving i) sequences of chemical equations for elementary steps, ii) reaction energy profiles, directly representing computed energies against some reaction coordinate, and iii) reaction networks (specifically, Chemical Reaction Networks or CRNs) highlighting the connectivity between the different species participating in the mechanism (Figure 2.1). Along this section, we will examine the strengths and weaknesses of the three approaches: nevertheless, we shall recall that all these representations focus on the same core principles (e.g. element balance, electroneutrality...) and can be used and switched depending on the specific system which is the object of study.

Starting from the sequences of chemical equations, their major advantage is their clarity and understandability: balanced chemical reactions are one of the most deeply ingrained principles of Chemistry, thus making this kind of representation really accessible for almost every chemist. Moreover, it is

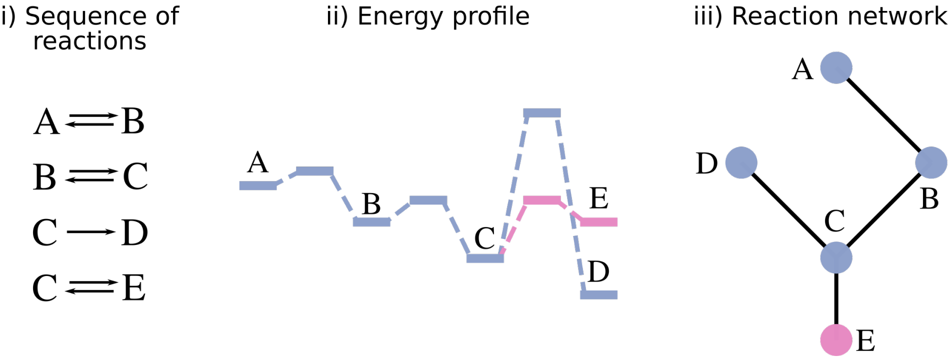


Figure 2.1: Comparison of the reaction sequence (left), energy profile (center) and reaction network (right) representations for a simple reaction mechanism including the overall transformations $A \rightarrow D$ and $A \rightarrow E$.

easy to convey additional information beginning from the basic "reactants / arrow / products" structure which is fundamental to chemical equations, including, for example, the energies of the involved states, additives or additional reagents, and so on. Nevertheless, this kind of description can easily get messy for complex multi-step processes that involve many elementary reactions, and even more when the corresponding states are interlinked and participate in several of the reactions of the set. Overall, these "plain" chemical reactions are unbeatable for bigger-picture schemes such as synthetic transformations (e.g., we may summarize the transformations in Figure 2.1 with the reaction $A \xrightarrow{\text{[Cat]},\text{T}} D + E$), but are much less adequate in mechanistic studies.

On the other hand, reaction energy profiles are particularly suited to computational chemistry, as they provide an immediate idea on the energy differences along multi-step reaction schemes. This allows to assess, at a glance, the more energetically demanding steps and the overall thermodynamic feasibility of the represented process. Profiles are often accompanied by labels or molecular structures tagging each of the steps, integrating composition information, just as reaction sequences (or reaction networks) might be accompanied by the corresponding energy values. Nevertheless, just like reaction sequences, energy profiles encounter issues when describing intertwined reaction mechanisms involving numerous side reactions or cross-transformations between species, as they assume a *linear* sequence of intermediates and transition states that does not provide a proper description in many situations.

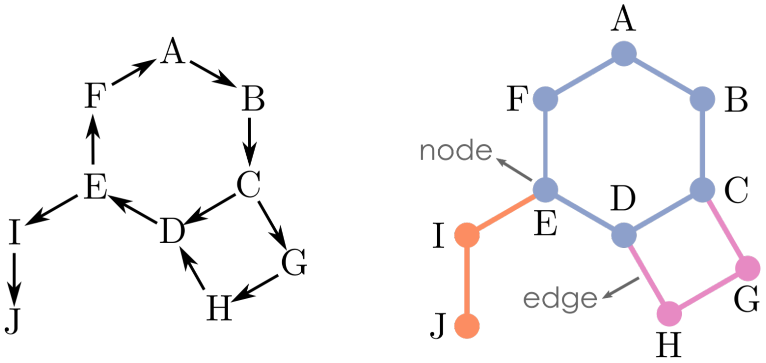


Figure 2.2: Representation of a non-trivial catalytic cycle from elementary reactions (left) and as a graph (right).

To properly describe this kind of intricate mechanisms, it is necessary to resort to *reaction networks*, where the main representational focus is the interconnectivity between the states happening throughout the mechanism. A classic example of a reaction network could be a catalytic cycle, where the elementary reactions from the initial state of the catalyst to its regeneration are depicted in a cyclic fashion (left part of Figure 2.2). Once the sequence of reactions has been reorganized in this way, the analogy with *graphs* is immediate: the intermediates (labeled with letters) correspond to the *nodes* of the graph, and the transformations that interlink them (reaction arrows), to the *edges*. Moreover, from the viewpoint of computational chemistry, the edges can be mapped to the transition states governing the transformations encoded in each elementary step, thus assigning chemical structures to the two types of entities in the graph. Through this representation, it is trivial to understand and manage caveats such as the off-cycle branch $E \rightarrow I \rightarrow J$ or the alternate path $C \rightarrow G \rightarrow H \rightarrow D$ that could be overlooked through reaction sequences and would be hard to depict through energy profiles. Although we have considered a catalytic cycle to exemplify this introduction of graphs to handle reaction networks, this representation can also be used for non-cyclic networks, as shown on the right of Figure 2.1.

Another important point that shall be noted is that in Figures 2.1 and 2.2 we have proposed graphs without any kind of direction arrow. We will delve deeper on types of graphs (and other aspects of Graph Theory) along Section 2.2, and on further considerations of directionality on graph CRNs for computational chemistry on Chapters IV and V. As of now, we will just state that both directed and non-directed graphs can be used to describe CRNs, with the choice of one or the other approach depending on the specific case and subfield. In the context of computational chemistry, undirected graphs provide a simpler solution where the chemical flow of the system can be determined *a posteriori* through the thermodynamics and kinetics encoded in the energies of the different states on the network.

2.2 Graph theory and reaction networks

Graphs, which were briefly mentioned while introducing the notion of a reaction network, are mathematical objects composed by a set $V(G)$ of **nodes** or vertices and a set $E(G)$ of **edges** that connect two nodes in the set [39–41]. These objects are often represented by drawings where nodes are placed at some point of the plane with lines depicting the edges (Figure 2.3). The aim of this section is to provide only a brief introduction to some core aspects of Graph Theory that are important to properly discuss the work compiled on this Thesis. Plenty of specific and more detailed literature on the matter is available, owing to the large variety of fields in which graph-based modeling is useful (physical and social sciences, engineering, computing...).

In the most general case, an edge may connect a vertex to itself (self-loop) and there might be several edges connecting a single pair of nodes (multi-edge). However, the subclass of graphs where neither loops nor multiple edges are allowed, denominated *simple* graphs, is of particular importance for many modeling situations. Another important distinction

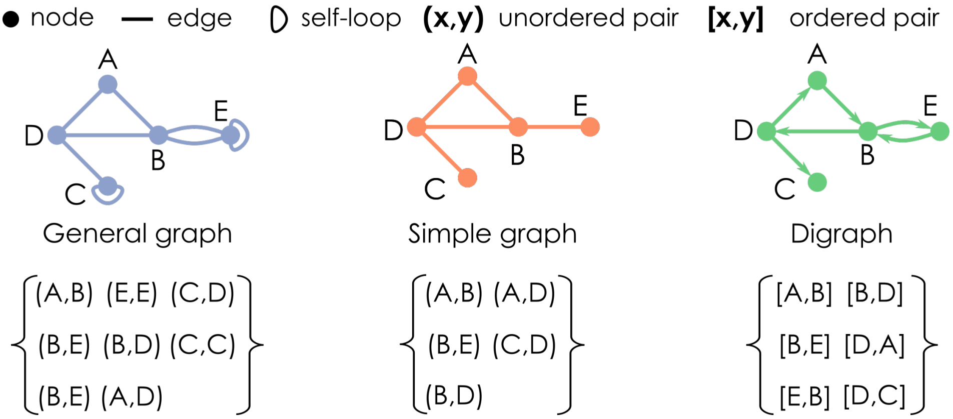
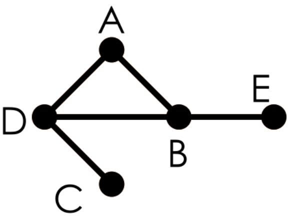


Figure 2.3: Representation of a general graph containing two self-loops (left), a simple graph (middle) and a digraph (right), all of them with a common set of nodes $V(G) = A, B, C, D, E$. Below each graph, the corresponding node sets $E(G)$ are shown, with nodes being in alphabetic order for undirected graphs (unordered pairs) and directed for the digraph (ordered pairs).

depending on the definition of the edge set is the **directedness** of the graph, that will be a directed graph or *digraph* if these pairs are ordered (with a directed edge being named an *arc*), and undirected otherwise. Figure 2.3 includes graphical examples of general, simple and directed graphs with their underlying edge sets, for clarification.

Along this Thesis, we will refer to undirected, simple graphs unless otherwise stated, as they provide a simple but accurate representation for chemical reaction networks (as mentioned on Section 2.1), particularly in terms of computational chemistry. In this way, no *a priori* assignment of how a given network will be traversed is necessary to define it, allowing further applications (such as these developed in Chapters III - V) to decide it from the chemical information it encodes.

A graph can be not only defined as a set of lines and points, but also as a matrix, which results a convenient representation in terms of efficient manipulation and storage. The *adjacency matrix* **A(G)** of a simple graph with **n** nodes and **m** edges is a **n** x **n** matrix where each element $a_{i,j}$ takes the value 1 if there is an edge between the nodes (i,j), which are considered adjacent, and zero elsewhere.

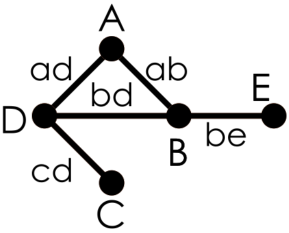


$$\mathbf{A}(\mathbf{G}) = \begin{array}{cc} \begin{array}{ccccc} A & B & C & D & E \end{array} & \\ \begin{pmatrix} 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix} & \begin{array}{c} A \\ B \\ C \\ D \\ E \end{array} \end{array}$$

Figure 2.4: Example graph, with labeled nodes (left) and its adjacency matrix (right),

For non-simple graphs, $a_{i,j}$ would be the unrestricted number of edges between the pair (i,j), and the diagonal elements may be distinct from zero in the case of self-loops. For reference, the adjacency matrix for the simple graph in Figure 2.3 is represented in Figure 2.4. It is important to recall that every adjacency matrix depends on a given node ordering that shall be also specified to properly define the graph.

Apart from the node-centric adjacency matrix, it is also possible to define the *incidence matrix* $\mathbf{M(G)}$, a **n** x **m** matrix where an element $m_{i,j}$ is 1 when the edge **j** is linked (or incident) to the node **i**, and 0 elsewhere (Figure 2.5).



$$\mathbf{M}(\mathbf{G}) = \begin{array}{cc} \begin{array}{ccccc} ab & ad & bd & be & cd \end{array} & \\ \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix} & \begin{array}{c} A \\ B \\ C \\ D \\ E \end{array} \end{array}$$

Figure 2.5: Example graph, with labeled nodes and edges (left) and its incidence matrix (right).

Adjacency and incidence determine the *degree* of a given node in the graph, which is the number of edges that are incident to it. The degree of a node allows to easily characterize isolated nodes ($deg(i) = 0$, not adjacent to any other vertex in the network) and end nodes ($deg(i) = 1$, adjacent to a single other node). From here, and related to these notions, another important general concept is that of *isomorphism*: two graphs G_1 and G_2 are isomorphic when there is a bijective relationship between their node sets that maps one to the other, as shown on Figure 2.6, preserving the edges.

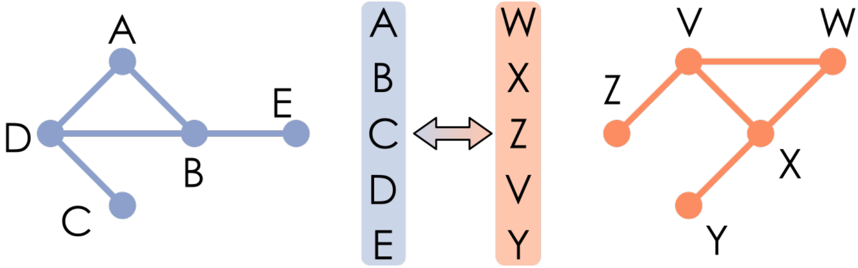


Figure 2.6: Graph isomorphism example, with the two isomorphic graphs at the sides and the node-to-node bijective relationship in the middle.

The corresponding adjacency and incidence matrices for two isomorphic graphs, provided that the node and edge orderings used to construct them are equivalent, will be equal: Equation 2.1 illustrates this for the two isomorphic graphs in Figure 2.6.

$$\begin{array}{cc} & \begin{array}{ccccc} A & B & C & D & E \end{array} \\ \begin{array}{c} A \\ B \\ C \\ D \\ E \end{array} & \begin{pmatrix} 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix} \end{array} = \begin{array}{cc} \begin{array}{ccccc} W & X & Z & V & Y \end{array} & \\ \begin{pmatrix} 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix} & \begin{array}{c} W \\ X \\ Z \\ V \\ Y \end{array} \end{array} \tag{2.1}$$

We have already mentioned the notion of going through, or *traversing*, a graph. The idea of encountering pathways to travel across a network directly connects with the idea of switching between reaction- or profile-based views of reactivity and graph-based reaction networks that was introduced in Section 2.1. Following graph terminology (Figure 2.7), a given sequence of adjacent nodes $n_0 \rightarrow n_a \rightarrow n_b \rightarrow n_c \rightarrow \cdots \rightarrow n_{-1}$ is called a *walk*, which becomes a *trail* when all traversed edges are distinct, and a *path* if all intermediate nodes are also distinct. Finally, a closed path where the initial and final vertices match ($n_0 \rightarrow n_{-1}$) is a *cycle*, a kind of path that is particularly common and relevant when discussing CRNs, and even more so for treating catalytic processes (see Chapter IV).

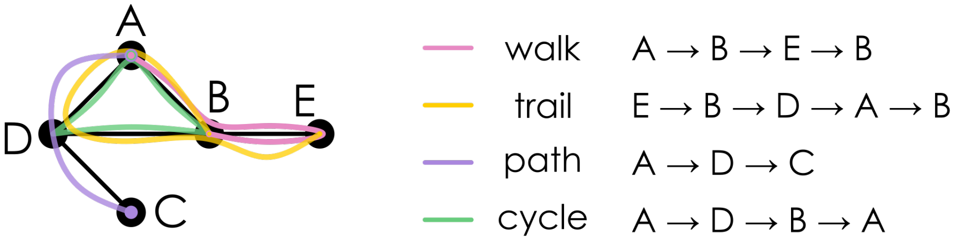


Figure 2.7: Examples of vertex sequences on a given graph including a walk (pink, repeating nodes and edges), a trail (yellow, unique edges but repeating nodes), a path (purple, unique nodes and edges) and a cycle (green, unique nodes and edges, closed path)

For a certain node pair (i,j), the shortest possible walk between the two nodes marks the *distance* $d(i,j)$ between them. For instance, for the graph in Figure 2.7, the distance between **D** and **E** is $d(D,E) = 2$ by considering the walk $D \rightarrow B \rightarrow E$ instead of the larger detour $D \rightarrow A \rightarrow B \rightarrow E$. In general finding such a walk is not a straightforward task, particularly with graphs of increasing size and complexity, and multiple algorithms have been developed to tackle this task (i.e., Dijkstra's algorithm [42]).

From there, we should also introduce the idea of *connectedness*: a graph is connected when there is a walk for every possible pair of nodes (i,j) in the network and thus every node is reachable from every other node. In any other case, there would be a disconnected graph that can be expressed as the sum of multiple individual connected graphs, named *components* (Figure 2.8)

The components of the disconnected graph on the right of Figure 2.8 illustrate two more graph types: an isolated node (in yellow) and a *tree* (in pink). Trees are connected graphs not containing any cycle: because of this, several problems that can be complex for graphs in general become trivial for trees, such as shortest path searches (as for a tree there is only a single

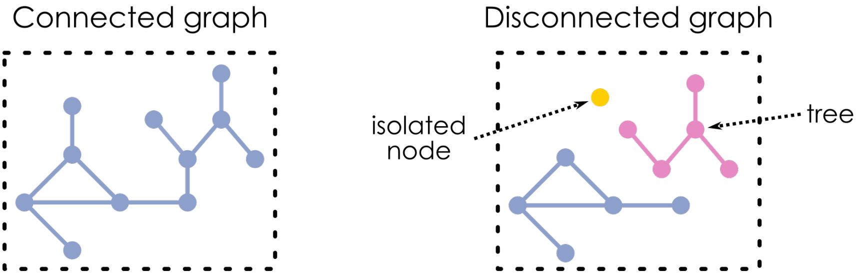


Figure 2.8: Examples of connected (left) and disconnected (right) graphs, with the three individual components of the disconnected graphs shown in different colors.

path connecting a pair of nodes). Consequently, trees are an important part of many algorithms and applications of Graph Theory.

From any given graph, it is possible to derive *subgraphs* whose node $V_{sub}(G)$ and edge $E_{sub}(G)$ sets are contained in the parent sets $V(G)$ and $E(G)$. The easiest manner to explain subgraph construction is by considering the deletion of nodes and/or edges from the parent graph (which would be the *difference* between the parent sets and the node and edge sets containing the vertices and edges to remove). Subgraph generation (Figure 2.9) will be a key part of Chapter III and especially Chapter IV, founded on this kind of network manipulation.

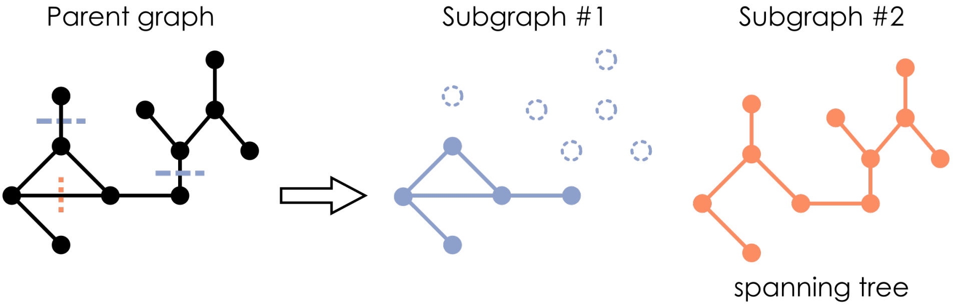


Figure 2.9: Examples of subgraph generation from a parent graph, removing nodes and edges (middle graph, blue) or edges only (right, orange).

The structure at the right part of Figure 2.9, in orange, corresponds to a *spanning tree*, which is defined as a fully connected, acyclic graph (fulfilling the definition of a tree) which spans all the nodes of the parent graph (and thus $V_{sub}(G) = V(G)$). The generation of spanning trees from a graph (tree growing) is a key part of numerous graph processing algorithms, and does also play an important role in the mathematical framework of the tool that we will present in Chapter IV to treat catalytic cycles, named gTOFfee.

In the context of algorithms related to the tree-growing strategy, and returning to the concept of path searches introduced along the section, it is worth discussing two core examples: the depth-first (DFS) and breadth-first (BFS) algorithms. Both of them take a graph and generate a spanning tree containing a set of paths across the network, following two opposite strategies regarding which incident edge and node will be added to the tree at every step (Figure 2.10). In DFS, the last added node is chosen as a source, thus beginning by going deeper into the graph. In contrast, BFS favors the nodes that were included first, therefore exploring first the complete neighborhood of the starting point.

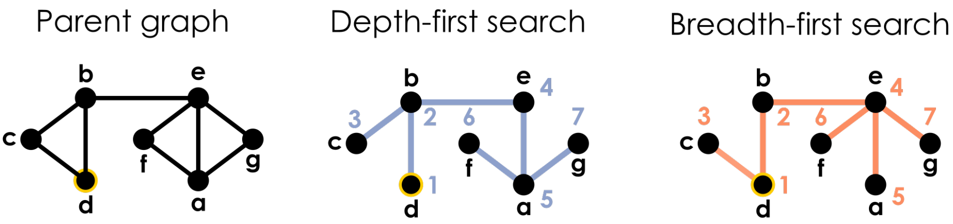


Figure 2.10: Spanning trees obtained by depth-first search (middle, blue) and breadth-first search (right, orange) for a given parent graph, adapted from Chapter 4 of Ref [41]. When two possible target nodes are tied for preference, lexicographical ordering on node names is considered as selection criterion.

Figure 2.10 illustrates how, although both DFS and BFS explore all nodes in the graph, the resulting spanning trees are indeed different. Starting from **d**, the first reached node is **b**, where the two strategies depart and reach **c** through two different edges, with DFS continuing from node **b** through the edge **bc** and BFS returning to **d**, which was added first to the queue, and going through **cd**. From there, **a** is reached via the sequence

$b \rightarrow e \rightarrow a$, generating another divergence: DFS goes on from this node and adds the edges **af** and **ag** and BFS comes back to **e** adding **ef** and **eg**. This approach serves as an example of both path search, with the branches of the spanning trees being different paths across the network, and subgraph generation, obtaining different trees with different traversal algorithms.

2.2.1 Graphs in Chemistry

The connection between Chemistry and Graph Theory is remarkably deep [43, 44], even if in principle most chemists do not explicitly employ the mathematical terminology behind Graph Theory to manage the graph-like entities appearing in the field. On the one hand, and as discussed in Section 2.1, graphs may be used to describe the reactivity of a chemical system (reaction network) [45], providing a very convenient representation for the setup of automation frameworks and processing techniques, as in the current work. A facet of graphs that we have not yet discussed, which is key for the management of CRNs, is the possibility of assigning additional attributes to nodes and edges, tracking not only labels and connectivity but also any kind of information about the system that is being modeled. Hence, recalling that CRNs match nodes with intermediates and edges with reactions (and transition states), we may assign to these graph elements any property that has been computed for the underlying chemical entities: geometry, energy, free energy, etc, so the CRN can be used to organize and access the part of the dataset generated by the calculations. This notion will be explored along all three following chapters, but especially in Chapter V, where we handle the transformation of reaction network graphs to *knowledge graphs* conforming proper databases.

Furthermore, graphs are also an ideal tool for molecular representations: at a conceptual level, molecules can be thought as collections of atoms that are interconnected through bonds, or, in other words, as collections of nodes and edges. Therefore, traditional schematic depictions of molecules would indeed be representations of *molecular graphs*, characterized by atom identities and their connectivity (Figure 2.11).

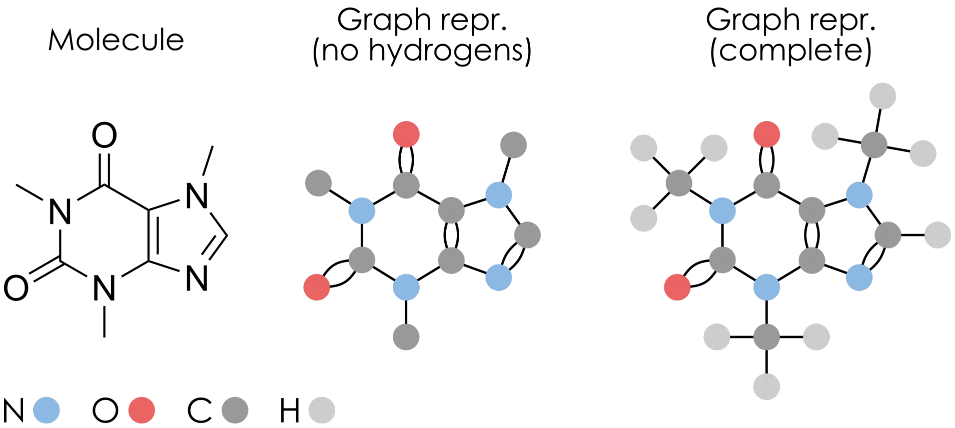


Figure 2.11: Caffeine molecule, in traditional 2D depiction (left), as a simplified graph without explicit hydrogens (middle) and as a complete graph including hydrogen atoms (right). Multiple bonds are depicted as multi-edges.

From this representation and as stated by A. Balaban [43], several traditional problems in Chemistry are, indeed, Graph Theory problems. For example, the determination of all constitutional isomers arising from a given molecular formula is nothing more than obtaining the unique ways in which a collection of atoms (nodes) can be connected, with the valence-related constraints limiting the connectivity of a given atom being related to the *degree* of the corresponding atom. From there, it is also possible to apply the graph framework to other related, although more complicated, aspects, such as valence isomerism, aromaticity... highlighting the depth of the linkage between graphs and fundamental Chemistry.

To take a look into how the constitutional isomerism question could be tackled through graphs, we will consider the example of determining the possible chemical structures following the molecular formula CH_5NO_2. From the "traditional" point of view, we should determine the number of unsaturations and cycles (also known as "hydrogen deficiencies") from the molecular formula [46], which in this case is $N_C - N_H/2 + N_N/2 + 1 = 0$, and determine, by hand, all the possible structures. Through Graph

Theory, the definition of the problem becomes a bit different: we need to obtain connected, non-isomorphic graphs from the set of 9 atomic nodes, fulfilling the degree constraints marked by the valence of each atom. To do this, a possible (although not exhaustive) solution would be to generate arbitrary graphs following the degree sequence determined by the valences of carbon, nitrogen, oxygen and hydrogen, and then filter them out to remove isomorphic graphs and non-connected entities (Figure 2.12).

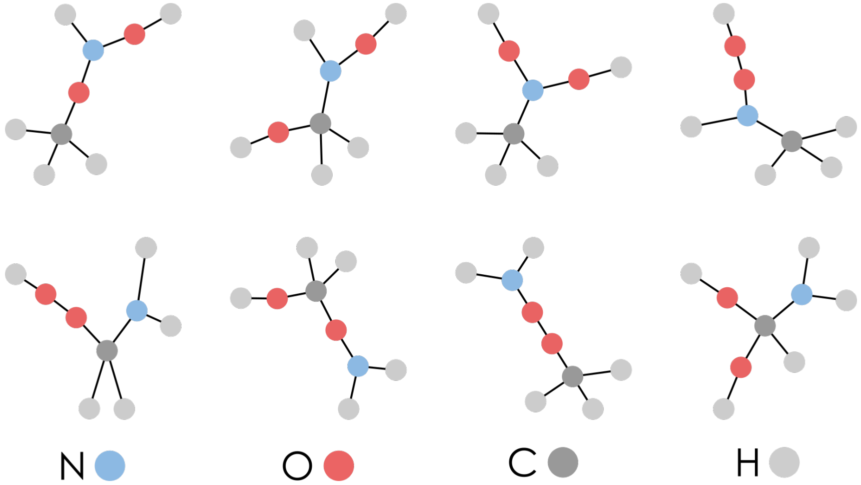


Figure 2.12: Molecular graph representations for the eight CH_5NO_2 isomers characterized through the exploration of N = 10000 possible graphs with the set of degree constraints corresponding to standard atomic valences.

While this approach does not ensure that every possible isomer is generated, the rapid convergence of the number of unique molecules $N_m = 8$ at $N_{test} \approx 100$, together with the inspection of the obtained chemical structures, seems to indicate a reasonable sampling. The collection includes all the unsaturated, neutral, usual-valence molecules that would have been proposed through chemical intuition, thus conveniently illustrating the relationship between graphs and basic chemistry.

Beyond this more fundamental chemistry, treating molecules as graphs has been a key part of numerous recent computational studies of chemical systems and materials, such as the development and application of chemical graph neural networks (GNNs) to drug discovery [47–49], the automated generation of reaction mechanisms and profiles [50], the prediction of adsorption energies over metal surfaces [51] or the detection of the elementary reactions involved in the speciation of complex metal oxides [52], among many other examples. Therefore, although this Thesis focuses on the use of graphs to model reaction networks and not individual molecules, the importance of molecular graphs is undeniable and essential to contextualize the profound relationship between Chemistry and Graph Theory.

Regarding computational chemistry, which processes and produces tridimensional geometries, it is important to decide a consistent criterion to assume whether two atoms are bonded, so the 3D structure and the molecular graph can be properly mapped. The direction in which the mapping is applied (either from the molecular geometry to the graph or viceversa) will depend, of course, on the specific problem that is being tackled. In general, obtaining the graph from the 3D structure is a more simple problem, allowing the immediate definition of the graph once a set of bonding rules has been chosen (distance thresholds, presence of Bond Critical Points according to the Atoms in Molecules (AIM) theory [53], etc). In contrast, embedding proper 3D structures from the connectivity of a graph requires additional information on the distances, angles and dihedrals expected from given atom combinations and, most often, also a preoptimization to refine the crude geometries obtained in the initial stage, as done in cheminformatics toolkits like RDKit [54] or OpenBabel [55].

Furthermore, there is a deep interest in transforming molecular graphs into molecular string representations: computer-friendly notations for molecules encoding the complete molecular structure graph as plain text. For instance, SMILES (Simplified Molecular Input Entry System) [56, 57], which is probably the most popular string representation, was originally defined as a language for encoding molecular graphs through a well-defined grammar and a set of symbols for atoms (nodes) and bonds (edges). Unlike molecular drawings, string-based descriptors can be directly processed by computers, and have much smaller memory requirements than mathematical representations such as adjacency or incidence matrices. Therefore, they have become an essential asset in cheminformatics, with their applications ranging from the normalization of chemical structures in databases [58, 59] to Deep Learning [60]. Alternative representations have been proposed along the years to improve the expressivity or performance on different tasks against SMILES, such as InChIs [61, 62], DeepSMILES [63], SELFIES [64] or TUCAN [65]. However, the field is still under active development, as several important limitations still exist to, for example, properly define organometallic complexes where valences are much fuzzier than in standard Organic Chemistry.

2.3 Coding in computational chemistry

Every computational science has its roots on providing the computing machine with adequate instructions to do its intended job: in other words, on *programming* the machine to solve the scientist's questions. The whole discipline of computational chemistry is founded on the development of the necessary approximations and simplifications required to solve the foundational equations from theoretical quantum chemistry within the limits of the available hardware: for instance, the plethora of methods and approximations that have been developed in order to approximate solutions to the Schrödinger equation for multi-electron systems. Throughout the continuous evolution of hardware in the last decades, exemplified by Moore's Law (Section 1.3), there has been a simultaneous development of the accompanying software, allowing the treatment of increasingly complex systems, not only in chemistry but also in any other areas related with intensive computation or, in general, the use of computers. In this context, countless programming languages have been proposed, developed and expanded, each one bringing their own advantages, disadvantages and idiosyncrasies to the table.

While there are many ways to classify this sheer variety of programming languages, a coarse but useful distinction is the one between the ones that are *compiled* and the ones that are *interpreted.* Roughly, compiled languages such as C or Fortran are much faster and more efficient, thus being adequate for computationally intensive tasks: e.g., almost every *ab initio* or DFT calculation software, requiring strong "number crushing" capabilities, will be compiled. To reach this efficiency, the instructions on the source language are translated to low-level machine instructions during the process of compilation. However, this intermediate process provokes a loss of flexibility: the addition of any modification to the source code implies a recompilation of the complete program, hampering testing and adaptation of the code to specific needs and forcing a strict input pipeline to be defined. On the other hand, *interpreted* languages like Java, R or Python do not end up directly in machine language, but are instead executed line by line by an *interpreter.* In this way, interpreted languages are much more flexible, allowing to modify the source code on-the-fly without an intermediate compilation stage, therefore providing a more dynamic and adaptable paradigm. Nevertheless, this comes at the cost of a slower performance and a less precise control of how machine resources are employed. In terms of computational chemistry, this trade-off between versatility and efficiency makes interpreted languages ideal for information post-processing, task automation or visualization, with the computationally intensive tasks being taken care by high-performing compiled codes.

Among interpreted languages, Python is becoming a kind of "de facto" standard for scientific applications, due to its clear, understandable syntax and its huge package ecosystem. Furthermore, the main disadvantage of Python, which is its quite lackluster performance for intensive computation, can be overcame by delegating this kind of tasks to packages based on compiled languages like C (e.g., NumPy), providing large speed boosts and permitting costlier calculations to be done in an efficient manner.

While for each of the projects developed along this Thesis we have employed different sets of specific packages that will be discussed in the corresponding sections, there is also a set of core libraries vertebrating scientific computing and data analysis in Python that are worth mentioning.

First of all, NumPy [66] provides multidimensional array structures that enable to carry out fast, efficient mathematical operations that deeply improve the performance of calculations compared to Python's own data structures. NumPy is indeed at the roots of most number-oriented applications of Python, being the basis for the rest of the so-called "scientific ecosystem". Then, SciPy [67] collects a large set of more specific mathematical algorithms (integration, linear algebra, and so on), physical constants and statistical analysis techniques, being another major stepping stone for further scientific computation. In terms of data visualization, MatPlotLib [68] permits the creation and customization of multiple kinds of bidimensional plots, permitting not only the production of static images but also the direct interaction with plots for on-the-fly exploratory analyses. Given the focus of this Thesis on graphs and reaction networks, it is also important to mention NetworkX [69], a library for graph creation and management which permits a simple, versatile access to all elements composing a graph that has been widely used across this work.

2.4 The semantic approach to data

The management and analysis of data is a complex problem and open question which nowadays is as active as ever, as we introduced in Chapter I. Indeed, previous sections in this methodology chapter were already focused, in one or other way, on the different ways in which we can organize information about chemical systems. For example, the comparison between linear reactions and reaction networks sums up as nothing more than the comparison of two paradigms on how to present the relationships between the entities of a single collection of data.

In data-driven approaches, the key point is not actually about how to store the information, but about developing sensible, shareable and scalable models for the target data. This kind of well-structured data provides novel manners to retrieve, explore and utilize the underlying knowledge for widely different purposes.

Among these more sophisticated (“smarter”) approaches to data management, we may highlight one that was introduced more than twenty years ago, quite before the “age of data” that we are living in nowadays: the *Semantic Web* proposed by Tim Berners-Lee [70]. This Semantic Web is an extension over the World Wide Web (WWW) aiming to add logic and structure to the underlying data, permitting the application of reasoning schemes and making complex inferences over the dataset. This approach to information relies on several different protocols, as exemplified by Figure 2.13.

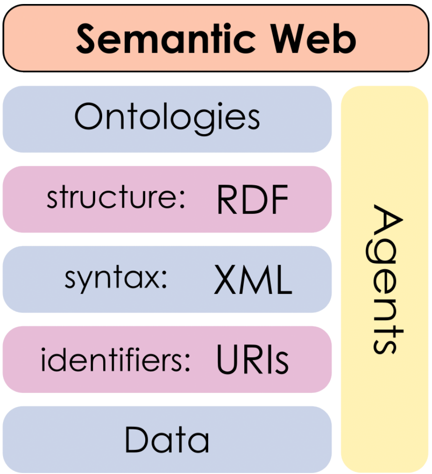


Figure 2.13: Semantic Web Stack illustration.

At the lowest level of the stack, we find the concept of Uniform Resource Identifiers [71] (URIs), allowing to provide global identifiers for the different entities in a dataset, providing a consistent notation scheme that may then be reused to connect different datasets containing common concepts. Indeed, the ubiquitous URLs (Uniform Resource Locator) employed to retrieve anything which is on the Web are a specific subtype of URI which can be retrieved in a web browser. URIs and information are then formatted as tags through the XML [72] (eXtended Markup Language) format, which gives **syntax** to the semantic documents by providing extensible and versatile tags that allow to express both data and metadata. Then, **structure** and **meaning** are added through the RDF [73] (Resource Description Framework) data model. In RDF, information (described through URIs and XML tags) is encoded as *triples* of the form subject - predicate - object: that is, we assert how a given resource (the subject) has a given property (the predicate), linking it to either another resource or a raw data value (the object). This model is very much in line with natural language, which is also roughly based in analogous subject - predicate - object constructions. A set of triples produces a non-relational database, where data is not organized in tables as in relational databases (e.g. SQL) but instead produces a **graph** through the connections between all resources in the dataset. Therefore, subjects and objects can be seen as the *nodes* of the underlying graph, while the predicates connecting them produce the *edges*, as shown in Figure 2.14.

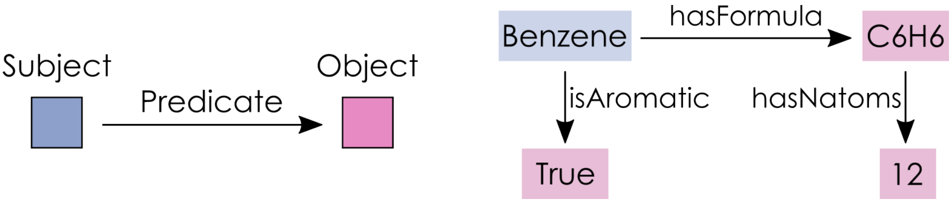


Figure 2.14: Example of the RDF data model, including graph representations for a generic triple (above) and a toy example of a RDF graph with three assertions about benzene.

This approach, compared to SQL-like databases, greatly simplifies the process of merging data from different sources, as the graph can be seamlessly grown by direct addition of new triples. Moreover, the universality of URIs does also contribute to this, identifying the matches between the nodes taken from different sources. In this way, it becomes possible to connect vastly different pieces of information with only a few common elements without the need to refactor anything on the data model, just by stating the existing relationships through a couple of new triples. However, this wide flexibility might collide with the initial idea of reasoning over the semantically-expressed datasets: if anything can be stated about any resource, how can we know what a given statement truly signifies? This issue is tackled by defining representational vocabularies for the elements that exist in a certain domain of knowledge: *ontologies* [74]. In few words, ontologies state **classes** characterizing the entities in the field and **properties** that express the relationships between them. Thus, there is an effective standardization of the area of knowledge, as common terms and structures are explicitly defined, allowing different communities to share and reutilize them, following the paradigms of Linked Data [75]. Writing an ontology, therefore, supposes a foundational effort on the organization of a given area of knowledge (either a specific one or a wider one), as it is necessary to capture the most important aspects of the field to define the appropriate classes and properties. Nevertheless, the flexibility of the data model allows to easily extend the ontologies, facilitating the switch from an individual to a collaborative effort, also in line with the principles of the Semantic Web.

It cannot be denied that the initial idea of a wide Semantic Web taking over the World Wide Web and providing complete coverage over the information in the Internet has not truly crystallized. Nevertheless, it has not disappeared either, supporting efforts like Wikidata [76], an ontology-organized collection of information which stores semantic data that then can be fed to other projects such as Wikipedia. Moreover, semantic-based data organization is also being used in more specialized domains, as shown by projects as the J-Park Simulator [77] for industry and engineering. In general, datasets that have gone through the whole semantic stack to be expressed in terms of an ontology are denoted as **knowledge graphs** (KG) or **knowledge bases**, where the specific entities categorized under a given class are referred to as individuals of the KG. The language of choice for ontologies and knowledge graphs is OWL [78] (Web Ontology Language), which is an ontology-oriented extension of plain RDF, although alternative formats have also been proposed (e.g. Open Biomedical Ontologies or OBO [79]).

Scientific data is an ideal target for semantic approaches, providing, in general, complex datasets and strong theoretical frameworks behind the dataset which can be readily translated to ontologies. While every specific field or subfield would require a specific ontology, both ontologies and their corresponding knowledge graphs may be bridged by stating equivalences between common elements, as mentioned before. Therefore, semantic approaches to scientific data would allow for a larger degree of interoperability, helping multidisciplinary projects to come to shape. As of now, ontologies have been widely adopted in biology and biomedicine [79–81], while for other fields such as Chemistry the adoption of semantic approaches is much less extended. An overview on some of the existing chemical ontologies will be provided on Chapter V, as a preface to the discussion on our efforts on treating reaction networks in a semantic manner through a novel ontology proposal.

2.5 The ioChem-BD database

Given that most of this methodology chapter has been devoted to strategies and approaches to the processing of data, it seems relevant to also consider how this data is stored, as the effective starting point for any kind of manipulation pipeline. In general, electronic structure codes produce heterogeneous and non-standard output files, where the way in which common properties are expressed can show wide differences across different software packages. Moreover, the non-standard nature of these outputs implies that changes may arise even between different versions of the same program, difficulting manual inspection by the user and breaking parsing workflows.

In this context, the ioChem-BD [82–84] platform provides a robust approach to overcome these issues, transforming these heterogeneous outputs into the structured and richly-tagged Chemical Markup Language [85–88] (CML) format and including them in a distributed database which greatly facilitates the retrieval and sharing of the corresponding information. Moreover, a web interface allows to access the most relevant aspects of the stored calculations, such as 3D molecular visualizations generated from geometries, energies, vibrational spectra..., both from users' own calculations or from results published by other researchers. This protocol, in the end, allows to make computational chemistry results open and in tune with the FAIR principles: Findable, Accessible, Interoperable and Recyclable [89].

Regarding the key transformation step, a set of templates (based on the JUMBOconverters library [90]) is selected depending on the input format, with the currently supported codes being Gaussian, ADF, VASP, GronOR, MOLCAS, MOPAC, ORCA, QuantumEspresso, Turbomole, Amber, GROMACS and LAMMPS. Depending on the format, either a single file or a collection of files is processed to generate the corresponding CML files, in compliance with the CompChem standard. The upload process can be triggered either through the web interface or in a programatic manner through a Linux shell client.

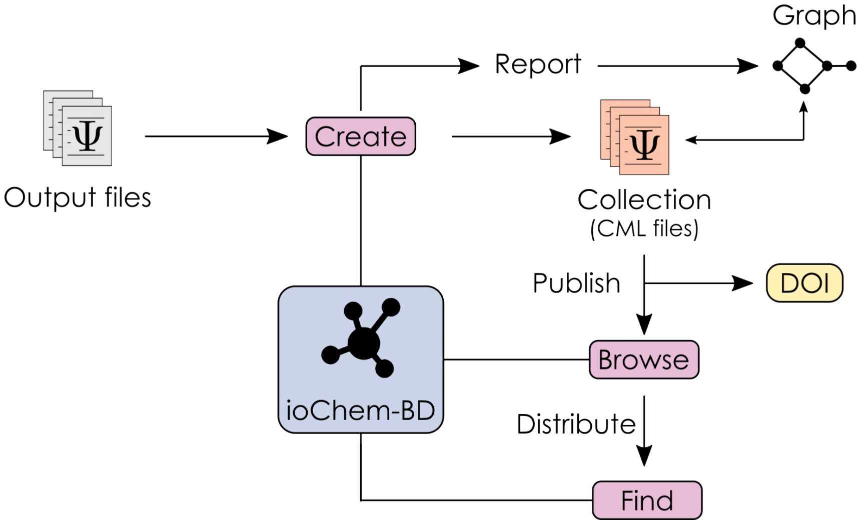


Figure 2.15: Schematic depiction of the core module structure of ioChem-BD and some of the main processes of interest for handling reaction mechanisms and networks.

Overall, the ioChem-BD service is comprised of three main modules: Find, Browse and Create (Figure 2.15). The first one, Find, provides a search engine over all data that has been made public across the different instances of the database. The last one, Create, handles the aforementioned transformation from individual outputs to CML files, allowing the user who generated the data to store the information in the database and work with it individually. Finally, the Browse module allows to publish the data collections defined in Create, making them accessible to other researchers by assigning a DOI to the dataset.

Another relevant functionality of the Create module is the possibility to build *reports* with additional information built on top of the individual calculations stored as collections. For instance, one type of report is the Reaction Energy Profile, in which the user inputs the sequences of steps happening in a given chemical reaction or set of reactions, in terms of the calculations stored in the database. From this information, it is possible to build the corresponding profiles automatically for different energy types (potential energy, enthalpy, Gibbs free energy...) and energy units ($\text{kcal}\cdot\text{mol}^{-1}$, hartree, eV...). Moreover, recalling the discussion on Section 2.1, energy profiles may be reorganized as reaction networks: therefore, this kind of reports eventually allow to express network topologies inside ioChem-BD. The functionality to build actual reaction network graphs from energy profiles was recently added to the platform, including the automated detection of transition state structures through the presence of negative frequencies to properly map nodes and edges in the resulting graph. The generated networks can be visualized directly in the web interface or downloaded in the standard DOT format for further processing.

As it will be discussed on the following chapters, the different projects undertaken along this Thesis have all been connected in some or other way to the ioChem-BD platform, encompassing new tools to add additional information to the raw calculations in the database (Chapter III), employing the graphs generated by ioChem-BD as input (Chapter IV) or constructing new data structures directly fed by ioChem-BD (Chapter V).

Chapter III

Taming automatically discovered networks: amk-tools

3.1 Automated mechanism discovery overview

Through the two previous chapters, we have already hinted at the sheer importance of automation in modern computational chemistry: not only to alleviate the daily workload of scientists, but to allow the exploration and understanding of systems that are too large or too complex to be treated in any other way. Thus, plenty of automation frameworks have been built for computational chemistry, tackling aspects that range from more specific issues such as conformational space exploration [91, 92] to customizable, general-purpose workflows that allow users to handle massive sets of calculations with minimal direct input [29].

The characterization of reaction mechanisms is a subfield where automated searches would be particularly useful, enabling a very profound investigation of all possible reactive pathways occurring for a given chemical system. In this manner, human-introduced mistakes such as the neglection of unexpected low-energy routes or the lack of consideration of competing side reactions can be avoided, attaining a more complete description. A perfect mechanism search engine would ensure to always find the best possible mechanism for a given computational setup (method, basis set...), obtaining a complete description of the reactive behavior of the system. However, and as it might be expected, this ideal concept is mostly a chimera, hindered by the enormous quantity of calculations it would require and by the non-triviality of the chemical space exploration problem: there is no obvious or unique algorithm to inspect how a molecule or collection of molecules may evolve. In this context, several concepts have been proposed along the years to develop suitable protocols for this kind of explorations, involving widely different philosophies with the common ground of ultimately "growing" a CRN [93]. For instance, the Artificial Force-Induced Reaction (AFIR) method from Maeda and Morokuma [94–97] allows to push two molecular fragments together (or to pull them apart) through the use of an artificial force, reshaping the PES of the system to cancel out energy barriers and obtain the so-called AFIR paths by minimizing the modified function. Moreover, the maximum-energy point of these paths provides a reasonable guess for transition state optimization, thus permitting an easy exploration of the major points of interest of the PES. An alternative proposal by Maeda and Ohno is the Anharmonic Downward Distortion Following (ADDF) method [98–100] (formerly named Scaled Hypersphere Search or SHS), which follows the distortion between the potential energy surface and the curve for a harmonic potential centered at the position of the minimum. The corresponding ADD curve built from the minimum, again, provides reasonable TS guesses that can be refined to readily locate true saddle points on the original PES. These two methods are implemented in the Global Reaction Route Mapping (GRRM) software [97, 101], which allows the automated application of these algorithms to find sequences of minima and transition states to characterize large sections of the overall PES.

Another approach to the automated mechanism search problem involves the use of molecular dynamics simulations to sample the PES of the target system. Given that the objective is to detect reactive events, these MD simulations shall not employ traditional force fields (molecular mechanics), which cannot modify atom connectivity, but instead methods that can describe the electronic structure of the system (ab initio, DFT, semiempirical), balancing accuracy and computational cost depending on the application. However, the main drawback of employing MD for reactivity studies is that chemical reactions are *infrequent* events that would require very long simulation times to be observed. As this limits the applicability of MD to sample single reactive events, plain direct dynamics are rendered completely unfeasible for the characterization of mechanisms that are composed by plenty of possible events. Therefore, the application of MD simulations to reactive processes requires to resort to *accelerated* molecular dynamics, in which different strategies are employed to bias the dynamics and explore the chemical space effectively. Among these enhanced sampling techniques, in a very general sense, we may highlight two widely used approaches: metadynamics [102, 103] and umbrella sampling [104, 105].

Both techniques allow the MD simulation to explore the energy landscape of the system according to a set of coordinates or *collective variables* that describe the key atomic displacements for a given reaction. In terms of mechanism generation, however, the predefinition of collective variables could be problematic, and other accelerated dynamics techniques have been employed such as temperature-accelerated dynamics [106, 107], boxed MD [108, 109] or trajectory parallelization [110]. From there, Martínez-Núñez and coworkers developed *AutoMeKin* [111–113] (Automated Mechanism and Kinetics, formerly known as *tsscds*, Transition State Search Using Chemical Dynamics Simulations), which combines the use of accelerated MD with a structure guess/TS optimization protocol similar to that of GRRM. In this way, transition states appearing along the MD simulation are sampled, optimized and connected with their corresponding minima, eventually building a set of reaction paths which conforms a chemical reaction network. Other protocol also involving MD simulations is the *ab initio* nanoreactor by Martinez and coworkers [114], where direct ab initio molecular dynamics (AIMD) are accelerated by applying a virtual piston to compress the system and bring the reactive molecules together. Through this technique, it becomes possible to model situations where many reactants are involved, such as simulations of the Urey-Miller [115] experiment for the formation of organic moleculers from the atmospheric components of the early Earth.

Along this chapter, we will be focusing, specifically, on processing and treating the reaction networks generated by AutoMeKin. However, as we outlined before, the production of CRNs is the common goal of reaction mechanism discovery tools. Thus, in spite of the different formats in which the network itself and the accompanying electronic structure calculations are generated, the general strategy that we propose here could be adapted to other codes in a relatively easy manner. As a final note and for consistency with the nomenclature employed in the program and its related literature, throughout the chapter we will employ the abbreviation RXNet to refer to reaction networks, instead of CRN as in previous chapters.

3.1.1 Workflow of AutoMeKin

In very few words, AutoMeKin's main feature is the generation of reaction networks from a single molecular structure, exploring the different reaction paths that this initial substrate may undergo and eventually characterizing a set of feasible transition states connecting the structure with contiguous minima in the PES. Then, the new minima are used to start additional iterations of the protocol, growing the network as outlined in Figure 3.1 until a satisfactory chemical space coverage has been attained.

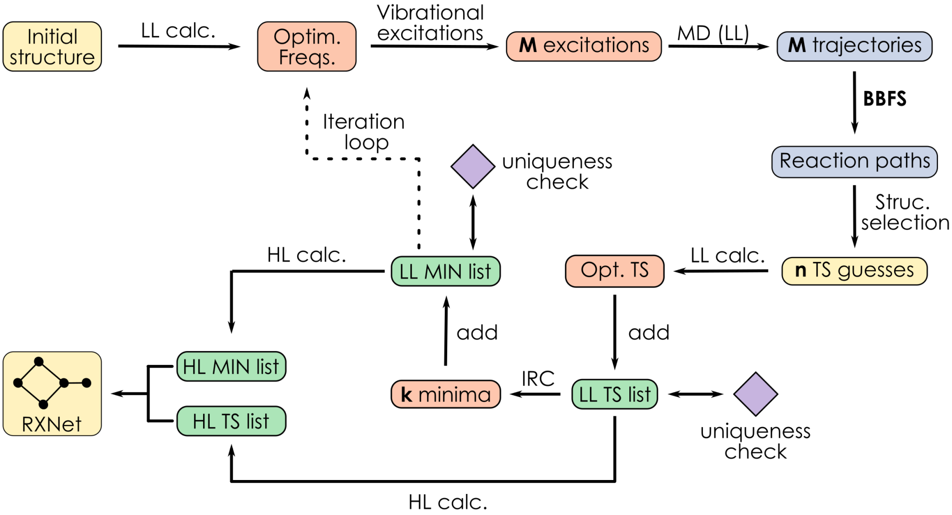


Figure 3.1: Schematic depiction of AutoMeKin workflow.

Giving some more detail, a key point of the workflow is the integration of two different levels of theory, with a lower one (**LL**) used to carry out initial preoptimizations and the molecular dynamics themselves, which should be a very affordable method (e.g. a semiempirical), and a higher one (**HL**) used to refine the obtained structures, (e.g. DFT or ab initio).

First of all, the proposed initial structure is optimized with the LL method, obtaining the corresponding frequencies. An ensemble of **M** vibrationally excited structures is then generated, so **M** MD simulations with the LL method will be run. A large degree of vibrational excitation is employed there to ensure the exploration of high-energy regions of the PES with short simulation times on the hundreds of picoseconds. Depending on the simulation conditions, either excitation energies (microcanonical NVE ensemble) or temperatures (canonical NVT ensemble) will be used.

Then, the corresponding trajectories are analyzed in order to extract the corresponding reaction paths, through the Bond Breakage Formation Search (BBFS) algorithm, which allows to detect changes in the distances between atoms and their neighbors inside a certain time window. Several reactive processes can be characterized with the BBFS algorithm, from simple dissociations to 3-center o 4-center elimination or isomerization processes where concerted atom displacements take place. The use of time windows instead of instantaneous time frames is precisely done to avoid problems with the identification of these concerted displacements, properly characterizing the reaction path.

From there, the point at which the distance change is first detected is stored as the *transition step* of the path, and is used for the following structure selection stage of the protocol. Here, not only the transition step is selected, but also its neighboring points (after and before), producing an ensemble of **n** TS candidates for LL optimization. This ensemble generation increases the chance of obtaining a valid saddle point after optimization, as the transition step itself might not be the ideal guess in all situations. Finally, TSs whose optimization is successful will be added to a list of TS structures at the LL level, which later on will be checked to avoid having repeated transition states. This is achieved by defining the molecular graphs of the computed structures, and checking their isomorphism by means of Social Permutation Invariant (SPRINT) coordinates, which have the main advantage of being robust to the permutation of equivalent atoms (e.g., hydrogens in a methyl group). From the obtained TSs, the corresponding minima that a given TS connects are characterized by Intrinsic Reaction Coordinate (IRC) calculations, to then be optimized at the LL level and passed to a list of minima, which is also the subject of a graph-based uniqueness check analogous to that of transition states. At this point, the process can be iterated, using all the minima collected in the first round for subsequent excitation - MD - TS optimization - IRC - minimum optimization passes. Of course, while more iterations imply a more detailed mechanistic description, they also increase the cost: the final choice of how many rounds of iteration shall be done depends mostly on the system being characterized and the available resources.

In any case, once the iterative process is finished, the final ensembles of minima and transition states will be recomputed with the HL method, obtaining more reliable geometries and energies for the involved structures. From there, the final reaction network encoding the mechanism that has been discovered can be built, storing the connectivity information generated through the algorithm.

Apart from the reaction network generation workflow discussed in the previous paragraphs, several new features and tools have been added to the program since its first development. For example, the mechanistic exploration has been improved by adding new accelerated MD approaches like the Boxed MD (BXDE) from Glowacki and coworkers [108, 109], the capability to treat Van der Waals association complexes [116] or even an alternative, non-MD-based chemical space search method, named ChemKnow and based on heuristic reactivity rules and graph transformations [113]. Moreover, the treatment of the produced RXNets has also been improved, analyzing graph properties or carrying out Kinetic Monte Carlo (KMC) simulations [117, 118] to solve the kinetics of the system and obtain the population of the different states of the network.

3.1.2 Understanding automatically generated networks

One main drawback of this kind of automated mechanism discovery platforms, shared with many other automation and high-throughput calculation strategies is the complexity of their output. Even with the parsing of the large quantity of the generated output files being handled by the automation program itself, the interpretation of complex networks containing many intermediates and transition states is not trivial. This follows what we stated in Chapter II when discussing the different depictions of reactivity, with CRNs (a.k.a. RXNets) being ideal to represent intricate mechanisms, but far from obvious to grasp.

Consequently, although the very detailed RXNets generated by automated mechanism discovery tools imply, in general, a leap forward in the description of chemical systems compared with manual, intuition-guided searches, the advance in the actual *comprehension* of the mechanism can actually be hindered by this complexity. This mismatch supposes one of the central points of this Thesis, being the main reason behind both this chapter and the upcoming Chapter IV, targetting, respectively, mechanism discovery and catalytic activity.

Here, we propose interactive visualization as a solution to alleviate this complexity, kind of "reconnecting" the more abstract network entity containing nodes and edges with the chemical entities that stand behind it. Thus, we developed the *amk-tools* library [119, 120] to first *filter* the large networks generated by AutoMeKin through criteria such as energy thresholds, the presence of certain species or fragments, or maximum lengths for target reaction pathways, and then generate interactive dashboards to visualize these networks. The dashboards provide an unified interface to explore the network, freely panning and zooming on its nodes and edges, looking up the energies, geometries and even vibrational normal modes of the species associated to each of them, and visualizing and filtering the corresponding free energy profiles. We believe this approach to provide a much clearer view of the chemistry that is encoded in a given RXNet, facilitating its interpretation and making this kind of methods more accessible to a wider community. Beyond the generation of these dashboards, the library also streamlines the automatic upload of the obtained RXNets to ioChem-BD, thus putting together the individual calculations generated and driven by the program and the underlying network structure, in consistency with the principles outlined in Chapters I and II that we will also revisit in Chapter V.

3.2 Program details

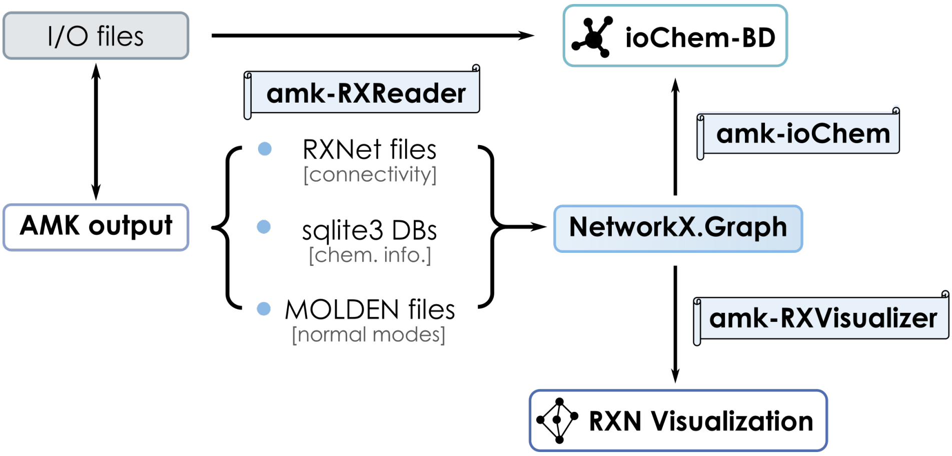


Figure 3.2: Schematic depiction of amk-tools workflow, comprised by the three main modules amk-RXReader (output parsing), amk-ioChem (automated upload of RXNets to ioChem-BD) and amk-RXVisualizer (generation of interactive visualizations).

As shown in Figure 3.2, our code is composed by three distinct modules, with *amk-RXReader* being in charge of i) parsing AutoMeKin output, including connectivity and calculation information, and ii) filtering it in order to generate the desired RXNet, *amk-RXVisualizer* for iii) generating of interactive visualization dashboards and *amk-ioChem* for iv) uploading the network and its calculations to ioChem-BD. We will provide details on how each of these four tasks is carried out, to give a clearer idea on how the library is designed and which are its capabilities.

3.2.1 Processing pipeline

Although the raw input and output files of the electronic structure program that is interfaced are also available, and will be targetted when dealing with the ioChem-BD upload process, the information of the reaction network is not directly extracted from there, but instead from pre-parsed files built along the mechanism discovery protocol (Figure 3.2). It is important to note that all the files we will discuss are available for both the LL and the HL calculations, allowing to create networks and visualizations at any of these levels.

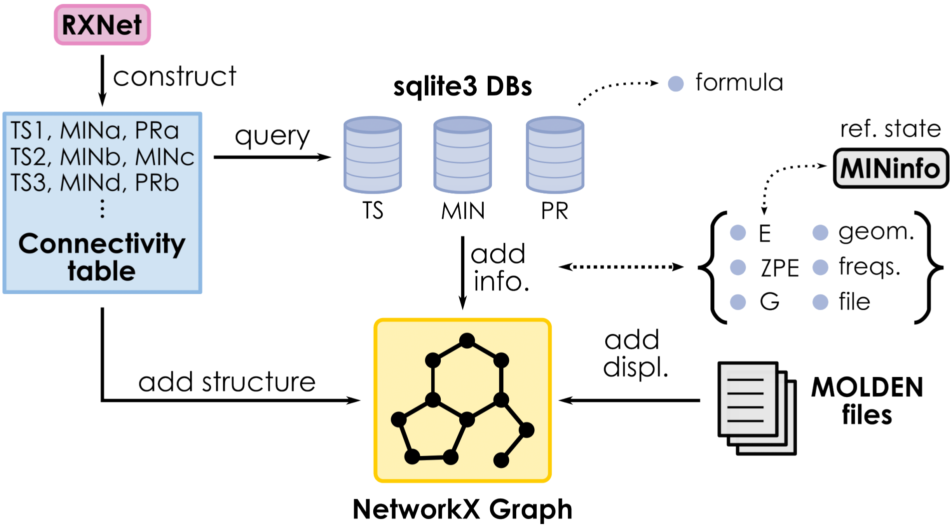


Figure 3.3: Schematic depiction of the parsing protocol of amk-RXReader.

The RXNet files provide graph connectivity, containing numbered transition states, their relative energies in kcal · mol^{-1} (against the lowest-lying minimum) and the reaction step they correspond to, indicating the connected minima and the reversibility or irreversibility of the process. This reversibility information is not employed at all: following the principles outlined in Section 2.1, networks are here treated as undirected graphs, not including any preassumption on the direction along which a given transformation shall occur. Several different RXNet files are produced, containing different degrees of detail on the target mechanism: from the complete network including all the discovered minima and saddle points, to a strongly simplified subgraph where only the paths deemed relevant by the KMC simulations are included.

The most relevant information about the individual intermediates and transition states (energies, vibrations, geometry...) is collected in databases, simplifying the access to the properties of each calculation and consequently the population of the final reaction network graph. Both RXNets and databases set a distinction between two types of minima: fragmented entities, which are labeled as products (PR) and non-fragmented species that are just named minima (MIN). Finally, the vibrational normal modes of minima and transition states that are required to embed normal mode animations in the visualization dashboards are not directly included in the DBs, but instead stored as independent files.

The parsing process (Figure 3.3) begins by processing the RXNet file of choice, storing a mapping between the indices of the transition states and the minima that they connect to obtain the core network connectivity. Then, the databases are queried to retrieve the energy, ZPE, geometry and frequencies for every TS and minimum. For products (fragmented minima), the stoichiometry (or "formula") of the fragmentation that has taken place is also extracted at this poing. When TS energy is absent, as it happens when employing the barrierless RXNet (which does not have proper TSs) the energy of the highest-lying intermediate it connects is employed instead, with all other fields being left blank. Then, relative energies are computed using the most stable minimum as reference state. In general, the energies reported by the current version of the code (which are all computed in this part of the workflow) are electronic energies with zero-point vibrational energy corrections. Through this iterative process over the connectivity table, there is a check for repeated steps where the same minima are joined by more than one transition state, storing only the lowest-energy TS. This situation might happen, for instance, when using the coarse-grained RXNet which collapses conformers resulting from different IRCs, allowing multiple transition states to arrive at the same minimum. Once all the connectivity table has been processed, a NetworkX Graph object is generated with data from the calculations stored as node and edge attributes. Optionally, it is also possible to go through the MOLDEN files containing normal mode displacements and assign them as additional element properties on the pregenerated graph.

3.2.2 Filtering pipeline

Although the RXNet selection on its own already provides some degree of filtering, especially if the *relevant* network containing only the kinetically important pathways is used, the *amk-RXReader* module provides additional tools to filter out unwanted sections of the reaction network graph and simplify the target visualization. This capability is deeply ingrained with the detection of reaction paths across the network, as one of the filtering constraints that is available is precisely the selection of a subgraph containing only paths with a given origin and/or end and of a certain maximum length. Thus, we will start the discussion of filtering by commenting on the path search process: recalling the representation interchangeability mentioned in Section 2.1, it is possible to go back and forth between exhaustive lists of linear paths and interconnected reaction networks. Therefore, it is possible to filter a network in terms of profiles and recover the corresponding subgraph later on.

In general, graph traversal is a non-trivial question in graph theory, as stated in Section 2.3. Two types of path search are available, depending on the specification of source and target nodes. First, it is possible to do a brute-force search of all pathways encoded in the network, specifying only the starting point. Currently, the available implementation of this search starts by determining *cyclic* pathways in the network, decomposing the graph to its cycle basis and transforming each cycle of this basis to a path. Then, any edges that are yet untraversed are added as two-node, one-edge branches to the profile set. In its current state, the brute-force search is not too useful as an exploratory tool, as it will likely generate many very short profiles that do not provide too much information on their own. However, the protocol is included to allow the exhaustive transformation of a RXNet to a complete profile set, which is required for further applications such as the ioChem-BD connection.

In contrast, if both source and target are specified, a NetworkX built-in path search is carried out, using a modified depth-first search algorithm which takes the maximum length of the path between the two nodes as an additional argument. When using this approach, it is possible to state several source and target nodes, with the possibility of only specifying a given fragment stoichiometry instead of a specific node name. In this way, the network can be effectively filtered to only include the paths of a certain length between a starting point of interest (likely, the energy minimum) and a set of products.

3.2.3 Visualization pipeline

The visualization module is based on the Bokeh library [121] which allows for an easy generation of data analysis dashboards bridging Python code with webpage-like services relying on HTML and JavaScript that can be accessed through a web browser. In this way, dashboards are generated as independent and cross-platform HTML files not requiring a live Python server running behind, allowing for easier sharing and collaboration.

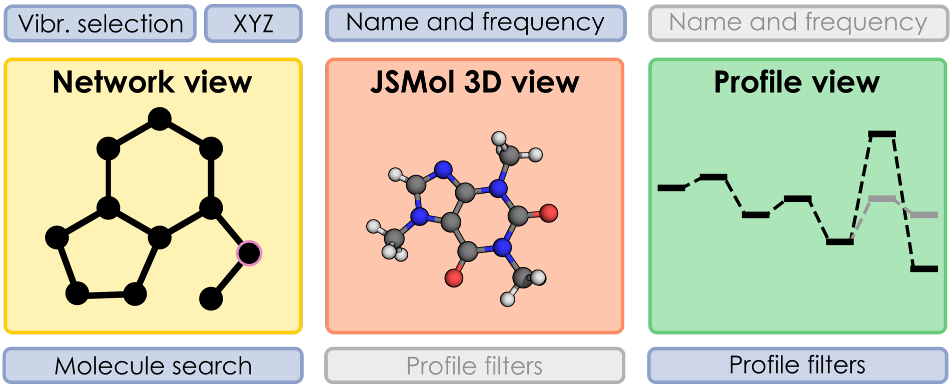


Figure 3.4: Schematic outline of the dashboards produced by *amk-RXVisualizer*, including view types and the major functionalities available for each one. Left, main network visualization, middle, 3D molecular model, right, energy profile mode.

The resulting dashboards contain the panels outlined in Figure 3.4, with the network view always present on the left side and the right side switching between the molecule and profile views. In the network panel, the RXNet parsed and filtered by amk-RXReader is presented, allowing to click on nodes and edges to select intermediates and transition states, freely zooming and moving around the reaction network to focus on specific sections for exploration or taking snapshots. The 3D models corresponding to the entities selected in this view will be shown in the molecule view. It is also possible to choose and load the different vibrational normal modes of a given molecule, showing the corresponding frequency values in cm^{-1}. Due to the philosophy of our code, all required information needs to be included explicitly in the HTML file, making it larger in size when a lot of data is introduced. To limit this size, it is possible to limit the number of vibrational modes that are processed when generating the visualization.

The molecular model view is powered by JSMol, an open-source JavaScript tool for the visualization of molecular structures which is widely used in browser-based chemistry tools such as ioChem-BD. While the own interface of the dashboard permits to carry out all the basic interactions with the molecule view, plenty of additional options are available through JSMol's own interface, enabling a deeper customization of the overall view.

Finally, it is also possible to visualize the reaction paths encoded in the network through the interactive profile view. While by default the complete set of available profiles is shown, is it possible to filter them according to either a maximum energy value (removing all profiles that have species exceeding the specified threshold) or the presence of the entities selected in the network graph. This post-filtering, combined with the previous network reduction, permits to easily explore and highlight different reaction routes encoded in a complex network, e.g. isolating the pathways leading to different product fragments, in a very simple and user-friendly manner.

3.2.4 ioChem-BD pipeline

While visualization dashboards can be directly shared to facilitate the direct exploration of a given reaction network, the data encoded inside them is not really prepared to be read or parsed, but instead to just feed the visualization. Moreover, only a subset of the information generated from the actual calculations is available on the dashboard. In contrast, ioChem-BD, introduced in Section 2.5, is a much more robust solution for data storage and sharing which can parse and host all the calculations generated through the automated mechanism characterization.

However, a direct mass upload of all calculation outputs will not be an adequate solution, as the chemically relevant information does not come from the isolated calculations but from the combination of these with the RXNet structure discovered along the mechanistic search. Therefore, an adequate protocol for connecting AutoMeKin and ioChem-BD will necessarily require to include this reaction network structure in the platform together with the calculations. This is achieved with the amk-ioChem module of amk-tools, which allows to go through the reaction network (as parsed by amk-RXReader) to generate a *report* in the platform containing connectivity information, linking it with the uploaded calculations to actually transfer the complete RXNet to the database.

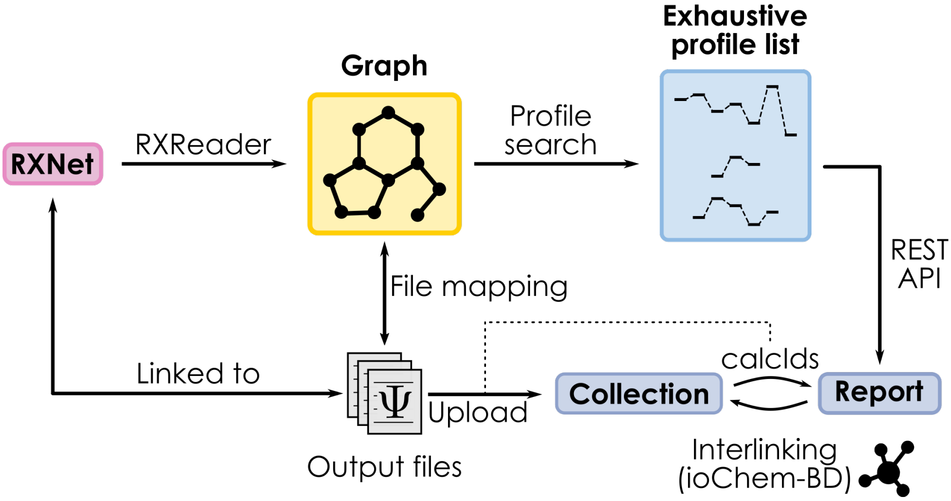


Figure 3.5: Scheme depicting *amk-ioChem* workflow.

While the workflow outlined in Figure 3.5 is quite simple, it is important to follow a certain order of steps to properly interlink the collection and the corresponding report.

First of all, the reaction network must be parsed to have a graph object, just like in the previous modules. Then, given that the definition of connectivity in ioChem-BD needs to be done by specifying energy profiles, it is necessary to transform the graph to a complete set of profiles including all connections in the RXNet. As explained when discussing filtering, this can be done either through a simple path search specifying sources and targets, or in a more automated fashion including each and every edge in the network. The choice of one or the other approach would depend on the specific target situation, making it possible to include only the paths of interest or to ensure to include everything that has been generated. Then, the next step is to map the stages in the profiles with the corresponding output files, eventually extracting two pìeces of information for the next step of the chain: a list of all found profiles and a mapping between calculation names and their corresponding input and output files.

The distinction between profile stages and individual molecules and its associated calculations is an important topic for properly handling reaction networks, which we will thoroughly comment on Chapter V as part of the formalization of CRNs tackled in that chapter. Stages, in general, do not correspond to individual molecular entities, but to *sets* of molecules that need to be considered altogether to achieve mass balance along the profile, so relative energies can be calculated. In the current context, most stages depend on an unique molecule and circumvent this issue, except for fragmented products, where each node does depend on multiple independent calculations (one per fragment).

To further clarify this aspect, Figure 3.6 shows the complete mapping sequence from a given node in the reaction network to the final identifiers generated after uploading the corresponding output files, highlighting how fragmented products involve multiple output files and therefore multiple items in the database. These items would eventually be linked to the corresponding fragmented stages through the *stoichiometry* information that is included with the profile definition.

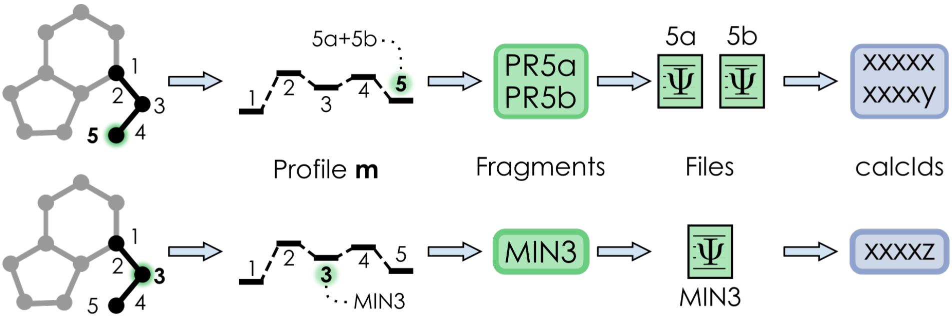


Figure 3.6: Example of the data pipeline in *amk-ioChem*, for either a fragmented product (above) or an intermediate (below).

Next, all calculations associated to the reaction network are automatically uploaded to ioChem-BD, pushing the input and output files. A key point of this part of the workflow is that the identifier of the calculation in the database is registered for every upload (as shown in Figures 3.5 and 3.6), so the items in the collection can be properly referenced. Finally, profile information is employed to define a set of reaction sequences that can be pushed to the database as reaction energy profiles inside a report (Section 2.5), to define network topology in the platform.

As of now, we have only described the different elements that comprise the amk-tools library and how they are connected to orchestrate the workflow in Figure 3.2. To give further clarification of the capabilities of our approach we will now present an example reaction network discovered by AutoMeKin, showcasing how our code streamlines its analysis, exploration and interpretation.

3.3 Application: the unimolecular decomposition of indole

Indole (C_8H_7N) is a 16-atom heterocyclic aromatic compound (Figure 3.7) including fused pyrrole and benzene rings which is present in widely different sources and has been shown to be relevant in multiple fields [122].

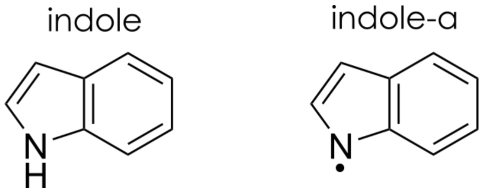


Figure 3.7: Indole molecule and indole-a radical resulting from hydrogen abstraction.

For example, indole is known to act as an intermediate on the biosynthetic route of the essential aminoacid tryptophan [123] and as a microbial intercellular signal [124]. It is also known to be an important nitrogen-containing component of coal tar [125, 126], from which it can be extracted and purified. Consequently, the decomposition routes of N-heterocyclic coal components such as indole are important in the more general study of coal combustion and pyrolysis, given that they may lead to the production of contaminant volatile nitrogen oxides. Because of this, computational studies unraveling the mechanisms for the pyrolytic decomposition of indole can be valuable for environmental chemistry. Recently, Liu and coworkers reported a DFT-based study on the decomposition channels leading to HCN and ammonia [127], whose previous exploration involved, mostly, experimental studies [128, 129]. However, the harsh conditions that coal pyrolysis implies make indole decomposition very likely to occur across a large number of possible pathways, providing an ideal case study for automating the mechanistic search instead on focusing on intuition-driven routes only.

Astrochemistry is another field in which indole is expected to play an important role, although the molecule itself has not yet been unambiguously detected in the interstellar medium (ISM). Due to its small size and chemical interest (simple, aromatic, N-bearing system), indole could take part in multiple important reaction routes in the ISM. This environment is characterized by very low molecule densities and temperatures, whose reactivity is governed by barrierless or quasi-barrierless processes. Because of this, and also taking into account the difficulties in confirming the presence of specific molecules in the ISM (employing techniques such as high-precision microwave spectroscopy), computational studies are essential for astrochemistry, proposing feasible reaction routes and aiding the spectral characterization. With all of this in mind, the automated exploration of the decomposition network of indole could also provide novel insights on possible formation channels, elucidating feasible chemical roles for this molecule inside the ISM. Assuming indole to be present in the ISM, its photolysis is already known to be a source of HCN via internal conversion from the excited to the ground state [130, 131]. Thus, a thorough mechanistic study could ascertain whether it can also lead to the isomeric HNC radical and therefore shed light into the yet unexplained branching ratio between HCN and HNC that has been found in the ISM [132–134].

Prompted by this interesting reactivity in pyrolytic and astrochemical processes, we explored the decomposition networks of both indole and the radical resulting from abstracting the hydrogen atom bonded to N (Figure 3.7) through AutoMeKin, using the resulting RXNets to test the visualization capabilities of amk-tools. In contrast with the general scheme that was outlined before (Figure 3.1), here we employed *three* different levels of theory for the overall study. The automated part of the protocol involved the semiempirical PM7 as the LL (or Level1) method and HF/3-21G as the HL (Level2) used for refinement and reoptimization of the LL network. From there, the pathways which we deemed relevant for our purposes, involving the formation of HCN, HNC, CN and NH_2 radicals for pyrolysis and a barrierless channel involving methylene radical for indole production in the ISM were recomputed at a higher (Level3) level of theory: CCSD(T)/aug-cc-pVTZ//M06-2X/MG3S.

Regarding simulation parameters, the complete LL workflow from Figure 3.1 was carried out for a total of 30 iterations, using $\mathbf{m} = \mathbf{500}$ MD trajectories of 0.5 ps per set. Saddle point guesses with imaginary frequencies below a 200i cm^{-1} threshold were discarded from the protocol. Furthermore, kinetic simulations were done using the RRKM theory to compute reaction rates and a KMC excitation energy of 250 kcal · mol^{-1}.

Before proceeding with the application of our library to analyze these results chemically, a couple of considerations on the six (Level1, Level2 and Level3 for indole and indole-a) reaction networks can be of interest (Table 3.1).

	Indole			Indole-a		
	Level1	Level2	Level3	Level1	Level2	Level3
Nodes	1973	1581	31	1360	566	13
Edges	2228	1878	34	1533	642	14

Table 3.1: Network sizes for the Level1, Level2 and Level3 networks computed for indole and indole-a decomposition.

For the two first levels, we will consider the coarse-grained networks including the barrierless channels, while the Level3, as commented before, contains a subset of fragmentation pathways leading to the small molecules and radicals of interest in pyrolysis and astrochemistry. We can see how the neutral indole network at the lowest level has almost 2000 nodes, which get reduced to around only 1600 when going from PM7 to the HF calculations. The huge leap between Level2 and Level3, keeping only 2% of the nodes at the CCSD(T)//M06-2X compound method, comes from two main factors: limiting the overall computational cost (regarding the costs of the Level3 method) and isolating the fragmentations of interest from the dataset from the numerous alternative reaction pathways that arise from indole.

	Orig.	L=8	L=7	L=6	L=5	L=4	L=8*	L=4*
Nodes	1581	410	392	348	251	129	119	45
Edges	1878	592	562	500	355	170	173	57
$\%_{red}$	100	26	25	22	16	8.1	7.5	2.8

Table 3.2: Network sizes for Level2 calculations for indole, including only pathways to selected fragments, with varying maximum accepted path lengths (from 8 to 4). Entries marked with an asterisk consider a threshold energy of 150.0 kcal · mol^{-1}, discarding paths involving structures with higher energies. $\%_{red}$ refers to the percentual proportion of nodes in the reduced graph against the parent Level2 network.

Starting from the Level2 network, we tested the path-related filtering capabilities of our code, considering different values for the maximum path length between the source node (the minimum-energy structure) and the desired fragments (CN, HNC/HCN, NH_2 and CH_2). To do this, we determined the number of nodes and edges remaining in the network depending on this cutoff path length (Table 3.2).

From there, we can see the clear effect that both path length and threshold energy filters have on the number of nodes and edges of the network. Even with a relatively large cutoff path length L=8, the number of intermediates in the RXNet is reduced to only a 26% of the original number if only this criterion is used, and to a 7.5% if the energy threshold is used too. Using shorter path lengths, the number of nodes keeps lowering, with the L=4 network having 129 (8.1%) or 45 (2.8%) nodes, without and with energy filtering. It is this manageable 45-node network which was eventually used for Level3 calculations, as the combination of the short path cutoff and energy threshold enforcement is expected to isolate only the most contributing reaction pathways.

Apart from the general numeric analysis outlined in Tables 3.1 and 3.2, we should also provide a visual idea on how the corresponding networks look at different degrees of filtering, as shown in Figure 3.8. While the complete network (top left) looks, as expected, completely untractable, the filtered graphs are much more manageable and thus chemically interpretable, especially under the interactive visualization framework we propose.

We may then consider an analogous analysis and representation for the RXNets of indole-a decomposition, considering different extents of filtering. As expected from the smaller initial number of nodes, the filtered indole-a RXNets (Table 3.3, Figure 3.9) are remarkably smaller than those for neutral indole: the L=4* network used as a reference for the Level3 calculations has only 16 nodes, achieving a 2.5% of the initial size which is very much in line with the reduction of the neutral indole RXNet.

From there on, we will focus on the Level3 networks for both indole and indole-a, analyzing the multiple fragmentation pathways that they encode

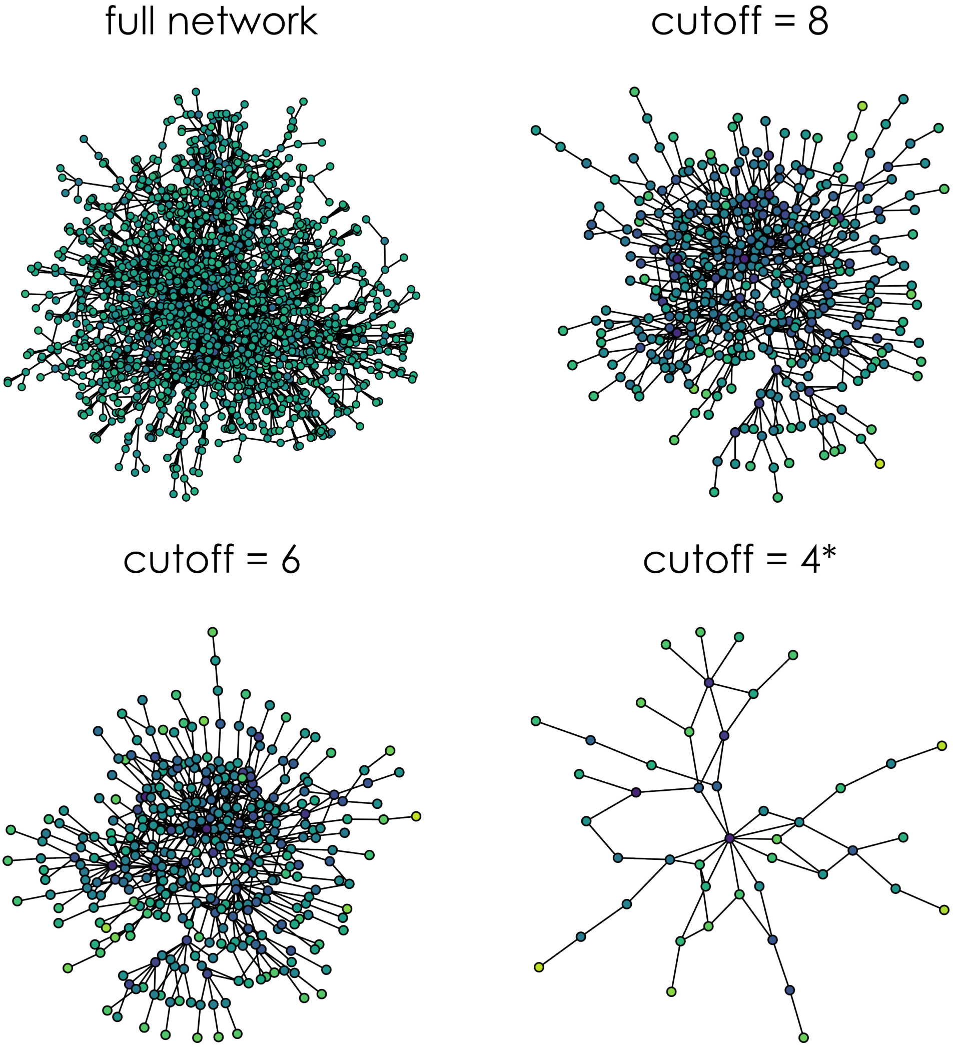


Figure 3.8: Level2 reaction networks for indole decomposition, including the full network (top left) and the filtered graphs with path length cutoffs L=8 (above, right), L=6 (below, left) and L=4 including a 150 $\text{kcal} \cdot \text{mol}^{-1}$ energy threshold (below, right)

and the interest of these in the contexts of pyrolysis and astrochemistry. Some of the intermediates and transition states appearing on the Level2 filtered networks used as reference (L=4 path cutoff, energy threshold of 150.0 $\text{kcal}{\cdot}\text{mol}^{-1}$) were not found to be proper minima or saddle points under

	Orig.	L=8	L=6	L=4	L=8*	L=4*
Nodes	566	146	85	51	57	16
Edges	642	180	96	56	69	17
$\%_{red}$	100	23	13	7.9	8.9	2.5

Table 3.3: Network sizes for Level2 calculations for indole-a, including only pathways to selected fragments, with varying maximum accepted path lengths (from 8 to 4). Entries marked with an asterisk consider a threshold energy of 150.0 kcal $\cdot$ mol^{-1}, discarding paths involving structures with higher energies. $\%_{red}$ refers to the percentual proportion of nodes in the reduced graph against the parent Level2 network.

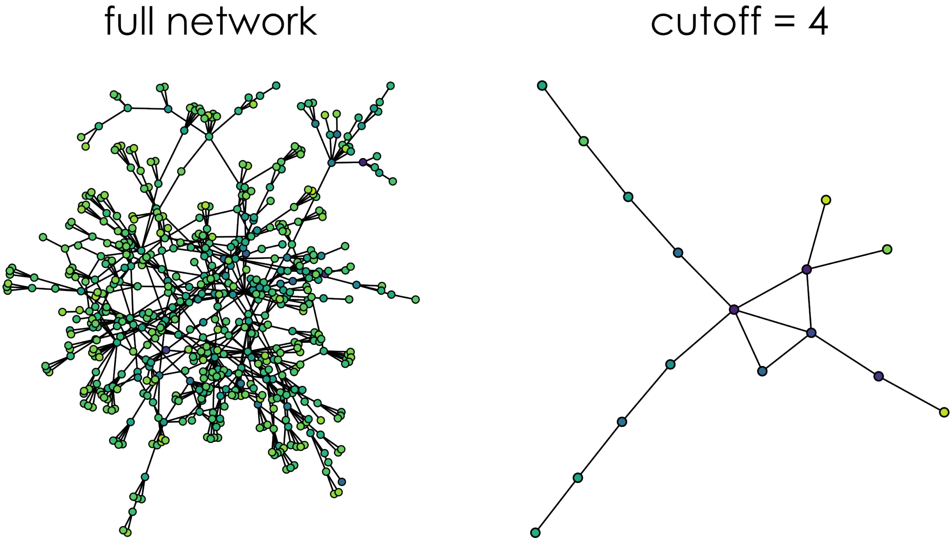


Figure 3.9: Level2 reaction networks for indole-a decomposition, including the full network (left) and the filtered graph with path length cutoff L=4 including a 150 kcal $\cdot$ mol^{-1} energy threshold (right).

the Level3 method, thus obtaining slightly different RXNets. Moreover, some channels which were below the threshold for Level2 calculations were higher in energy for Level3 results: despite being available, these will not be considered in the following reactivity discussion, which only regards these pathways below 150.0 kcal$\cdot$mol^{-1}. At this point, and before proceeding with the chemical analysis of the RXNets, we will present the actual interface of the dashboards produced for this system, illustrating the panels that were outlined in the previous section (Figure 3.4) to give a better idea on how this approach may assist with the navigation across the RXNet for indole (always within the limitations of the written format, only allowing static images to be presented). Nevertheless, interested readers may find the corresponding HTML dashboards for both indole and indole-a in the ioChem-BD repository [135].

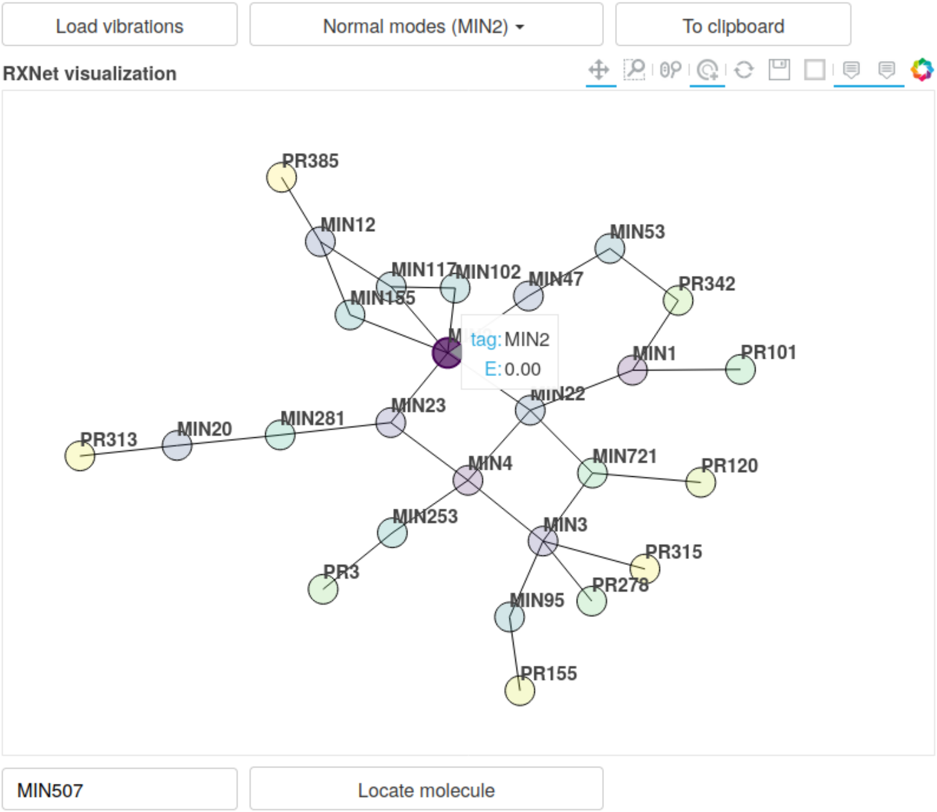


Figure 3.10: Network visualization panel for the Level3 RXNet for indole decomposition.

The network view (Figure 3.10), apart from the RXNet itself and the main controls summarized in Figure 3.4 presents a set of tools for navigating across the network, including the selection of nodes and edges or the depiction of their energies by hovering, as illustrated in the image for the species

MIN2.

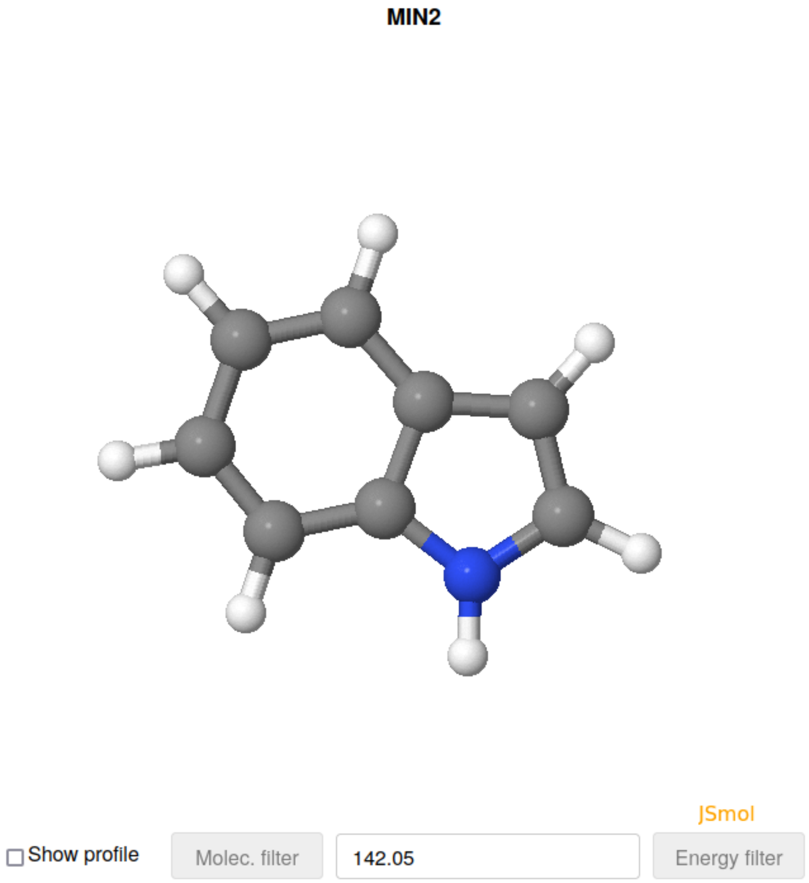


Figure 3.11: Molecule visualization panel, depicting the intermediate MIN2 for the Level3 RXNet for indole decomposition.

This view is always presented side-to-side with either the molecule (Figure 3.11) or the profile (Figure 3.12) visualizations, which can be switched through the “Show profile” checkbox appearing at the bottom right of the corresponding panels.

After presenting the toolkit, we will switch the focus to the analysis of the chemistry of indole decomposition. As stated at the beginning of the section, the pyrolysis of indole is known to be a source of nitrogen oxides (NO_x) upon burning coal tar, mainly through the oxidation of HCN

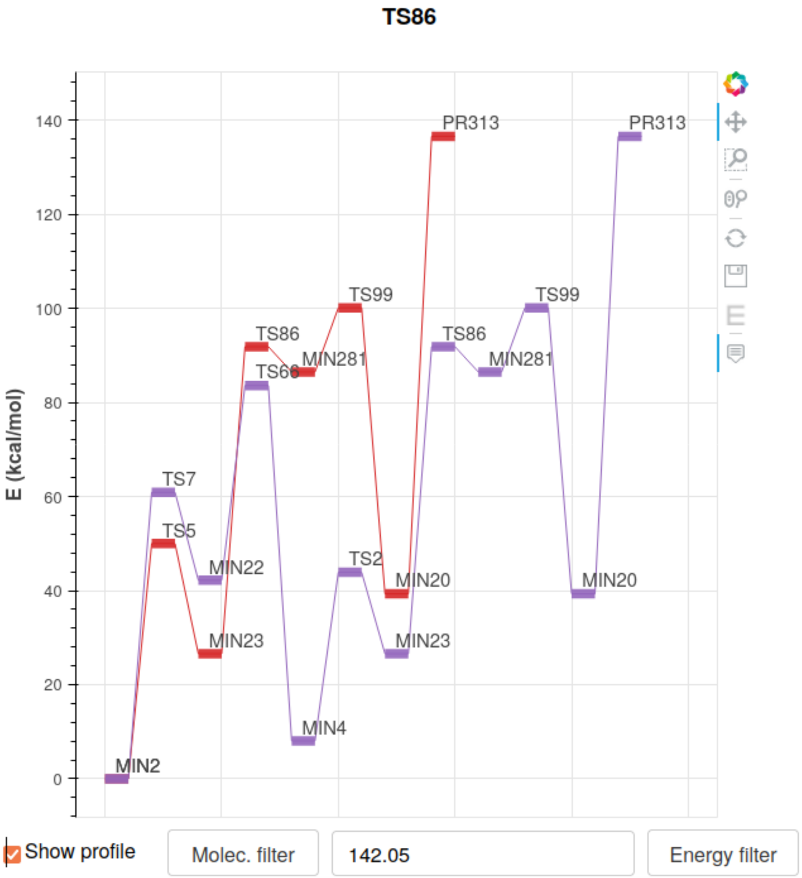


Figure 3.12: Profile visualization panel, showing profiles involving the transition state TS86, for the RXNet of indole decomposition.

and NH_3. To properly explore the formation of these two species, we shall consider not only the direct production routes, but also these leading to immediate radical precursors (CN and NH_2) and to the isomeric HNC molecule. To contextualize our results and justify the interest of applying automated discovery methods to this system, we will compare the obtained decomposition channels with these reported by Liu and coworkers [127] through a B3LYP/6-31G(d,p) traditional mechanistic search.

Starting with hydrogen cyanide formation and grouping together the channels from indole and from the indole-a radical, we determined a total

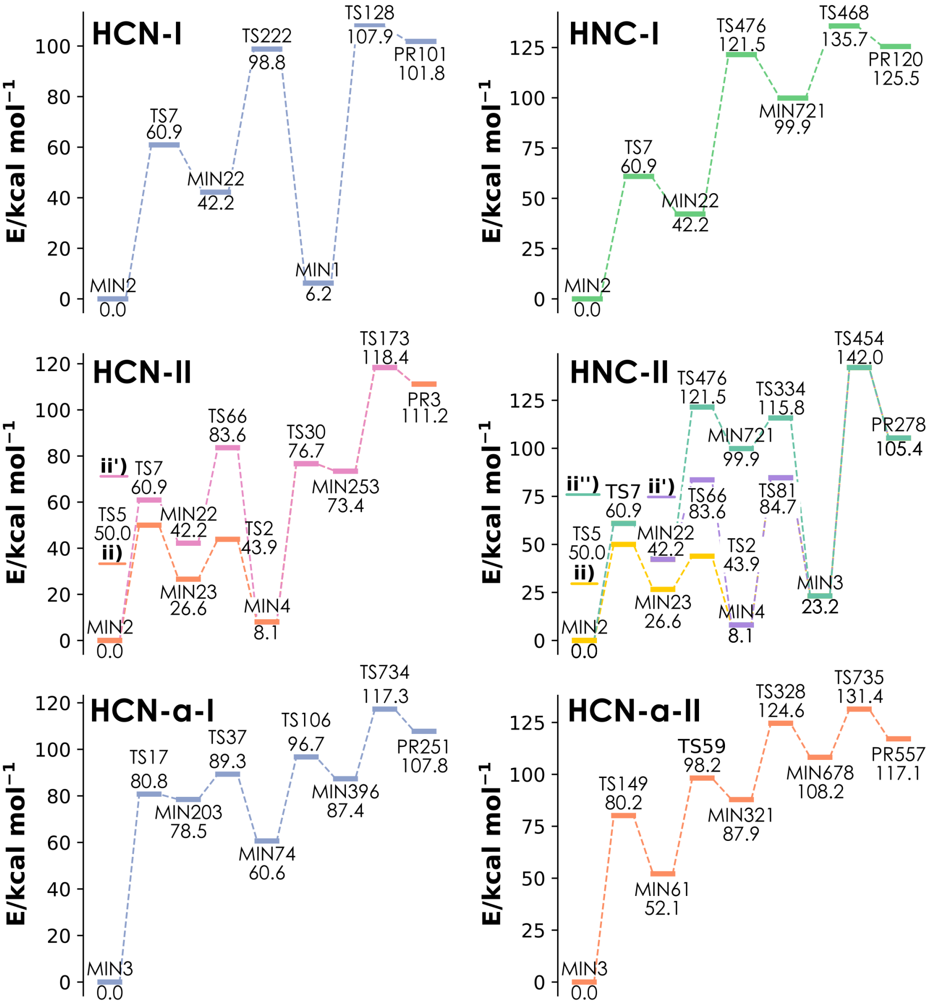


Figure 3.13: Energy profiles for indole (first and second row) and indole-a (bottom row) decomposition channels leading to HCN or HNC production. Reported energies are potential energies in kcal · mol^{-1}, including zero-point energy.

of five paths leading to HCN, four to HNC and nine to CN. In contrast, the previous DFT study did not include any route for the direct production of either HCN or HNC, only characterizing CN-forming channels. From these previously reported CN channels, two of them (labeled path-b and path-c in Ref. [127]) start from radicals that we are not considering in this study, abstracting hydrogen from the α and β positions of the pyrrolic ring, and thus are out of our current scope. On the other side, the other three reported pathways are effectively reproduced and identified by the automated protocol.

We will start the discussion with the nine paths leading to HCN and HNC radicals (Figure 3.13), grouping together all pathways sharing the same hydrocarbon fragment to obtain a total of *six* reaction channels. From this set of profiles, we see how HCN-I is the lowest-energy channel of the bunch and also one of the shortest, forming HCN in only three steps, thus involving one less stage than all other channels except for HNC-I. Although this simplicity should somehow favor this pathway against longer ones, the larger barriers of all these transformations together with the harsh conditions implied by pyrolysis will likely imply all these pathways to be intertwined and take place simultaneously, producing mixtures of the corresponding C_7H_6 (or C_7H_5 in the case of indole-a) side fragments. This situation is, indeed, a clear example of the duality that comes with representing complex reactive processes through either energy profiles or reaction networks. While the profiles in Figure 3.13 give an immediate idea of the accessibility of the proposed reactive channels, they do not make the high interconnectivity of the network clear enough. A more careful look on the profiles shows that many intermediates and transition states are shared between the different profiles, such as TS7 and MIN22 appearing in all four routes that start from indole, or the high similarity between the **(ii)** and **(ii')** pathways in HCN-II and HNC-II. If we isolate the part of the reaction network corresponding to these channels, effectively switching the representation from profiles to networks, we obtain the graphs in Figure 3.14.

This depiction makes the entwinment of the pathways that arise

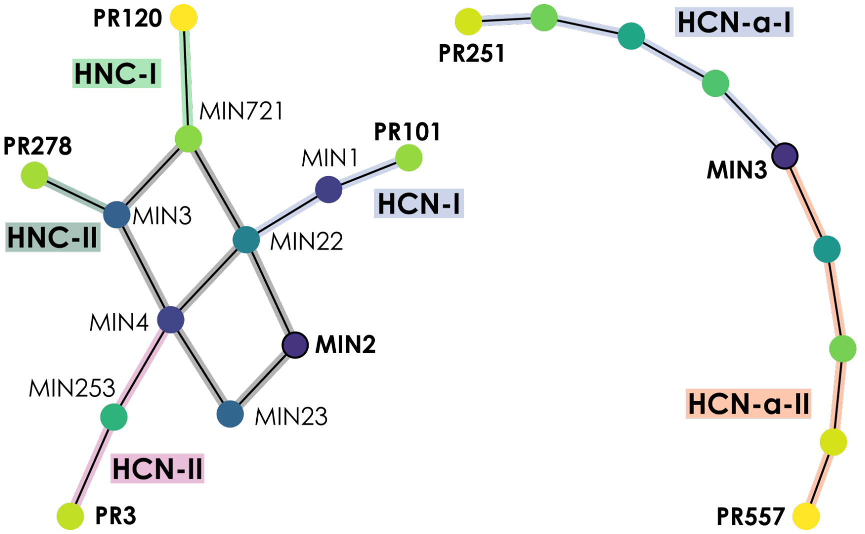


Figure 3.14: Reaction network representation for HCN- and HNC-producing pathways in indole (left) and indole-a (right) decomposition. Pathways are shaded with the same colors as in Figure 3.13, with the part in indole network that is shared by multiple channels being in grey.

from indole much more evident, highlighting, for instance, how every fragmentation route starting from MIN2 must necessarily go through either MIN22 or MIN23. Another aspect that becomes much more evident under this depiction is how the already formed fragments can isomerize by entering back in the reactive loop, which might be missed from the linear energy profiles. In the context of this chapter, the amk-tools framework simplifies the integration of the two approaches, leveraging the advantageous points that each kind of representation brings.

We may then do an analogous analysis for the pathways leading to CN radical (Figure 3.15), comparing them with the mechanism reported by Liu et al. Here, while all the pathways starting from indole-a end up at the same product (labeled PR193), they are depicted as distinct reaction channels due to their distinct path lengths.

The CN-I channel includes two of the pathways reported in the preceding DFT study [127], with CN-i corresponding to Path-1 and CN-i’ to Path-3.

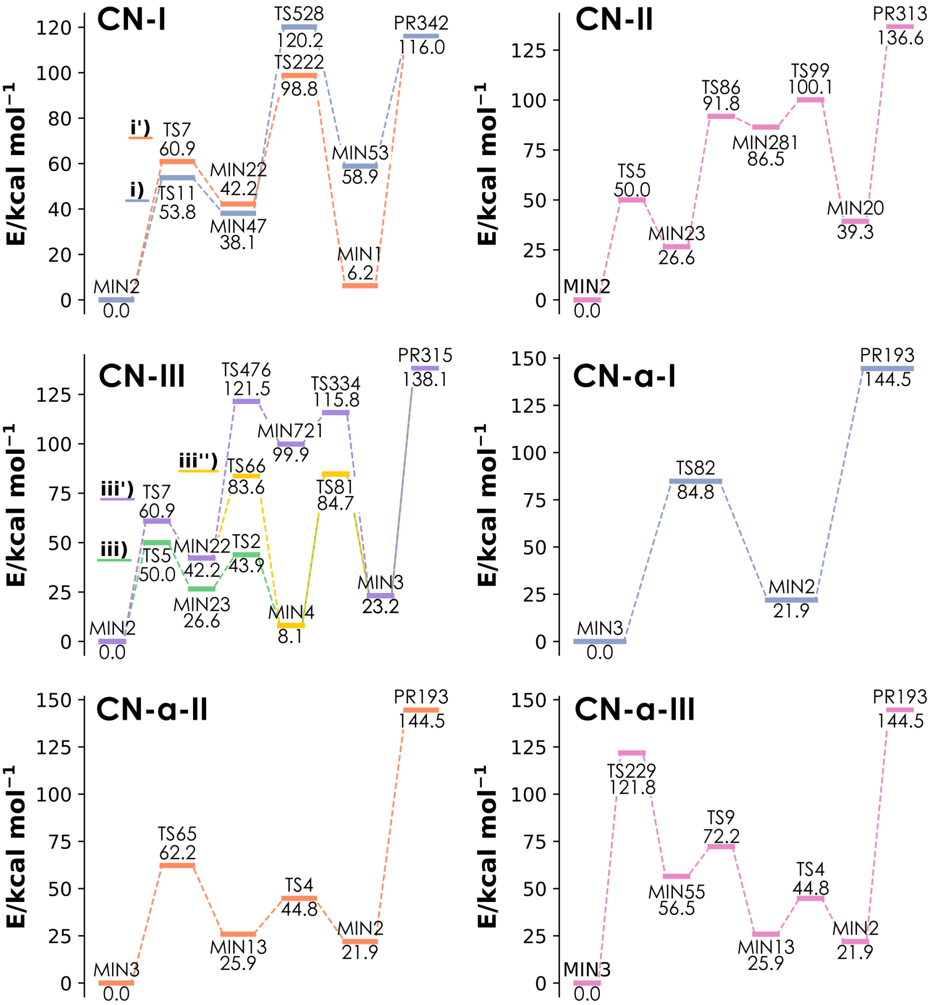


Figure 3.15: Energy profiles for indole (first row, middle row left) and indole-a (middle row right, bottom row) decomposition channels leading to CN production. Reported energies are potential energies in kcal · mol^{-1}, including zero-point energy.

The corresponding product (PR342) is in fact the most stable $CN + C_7H_n$ (n = 7,6 depending on the starting species) fragmentation of the set, thus enthalpically favoring the process. CN-II and CN-III include, respectively, one and three alternate pathways that were not found in the previous study, involving one more step than CN-I. As for indole-a decomposition, CN-a-II mimics Path-a from the previous study, with CN-a-I providing an unreported alternate and quite direct two-step pathway where hydrogen transfer and carbon - carbon bond breakage are concerted, which we expect to have an important degree of contribution to the network despite being higher in energy than CN-a-II. Finally, CN-a-III is another novel route, although due to being both longer and higher in energy than the other channels its contribution to the overall reactivity is expected to be quite low. If we compare now these channels with the routes for HCN/HNC formation (Figure 3.13) we see several shared transition states and intermediates, further stating the importance of intertwined routes to explain the reactivity of the system.

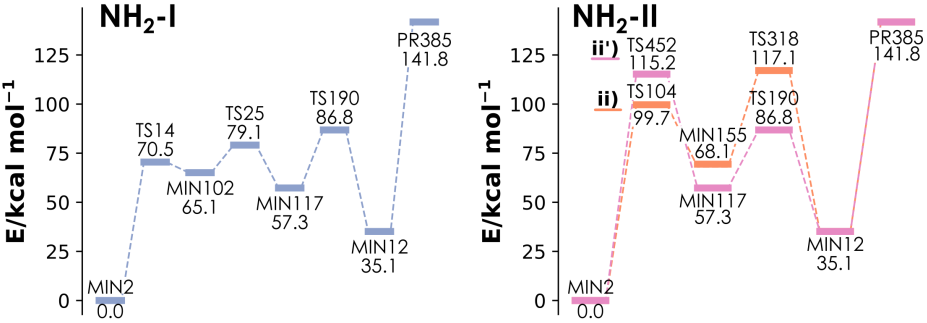


Figure 3.16: Energy profiles for amino radical formation from indole (middle row right, bottom row) decomposition channels leading to NH_2 production. Reported energies are potential energies in kcal · mol^{-1}, including zero-point energy.

Other important fragment that must be discussed to understand the pyrolytic decomposition of indole is the amino radical (NH_2), the major precursor for ammonia formation, given that no direct pathways to form

NH_3 were found neither in our study nor in the previous one. In comparison with CN and HCN/HNC formation, there are far fewer paths leading to amino radical formation, with only three viable routes from indole and zero from indole-a (Figure 3.16).

From these channels, NH_2-I approximately matches Path-2 from the previous study but including an additional step, with a ring disassembly process occuring through TS25 right before the formation of the NH_2 group in the intermediate MIN12 (2-ethynylaniline). In this way, the automated mechanism elucidation shows not only to be able to reproduce known pathways, but also to refine them when some aspects are undercharacterized. Moreover, Liu et al. proposed a second pathway (Path-4) that we found not to be a proper channel for amino radical production, but instead a self-loop connecting the intermediate MIN12 with itself. This aspect shows, yet again, the interest of applying automated strategies to revise and refine previous knowledge. Then, we identified a second reaction channel (NH_2-II) containing two unreported alternative pathways, higher in energy than NH_2-I but also one step shorter, which could also become relevant contributors to the general reactivity of the system. In general terms, the automated strategy is able to offer a richer description of indole pyrolysis in comparison with intuition-driven studies, reproducing and expanding the previously reported routes.

Regarding the possible roles of indole in astrochemical processes, the network exploration we are proposing could be an ideal tool to locate feasible barrierless gas-phase reactions leading to the production of indole in the ISM, in the more general context of the formation of interstellar complex organic molecules (iCOMs). Although formation mechanisms of iCOMs involve a variety of processes (photoactivated reactions, radical recombinations over grain surfaces...), gas-phase reactions have been proposed to play an important part in this chemistry. Nonetheless and as stated before, the harshly low temperatures of the ISM imply that these processes shall be barrierless, with the highest-energy TS happening along the reaction being always *below* the reactants.

Although our interest here is on the *formation* of indole from small neutral fragments, we may as well target the decomposition routes and fragments present in the RXNet and just look at them backwards, following the core notion of assigning the directionality of computed reaction networks *a posteriori*. A promising candidate reaction would be the combination of methylene radical (CH_2), which has already been detected in the ISM [136], and phenyl isocyanide (C_6H_5NC). While the latter has not been directly identified yet, the recent detection of its isomer benzonitrile C_6H_5CN [137] has prompted the interest on studying the isocyanide [138], whose formation from benzene and cyano radical has also been suggested to be feasible [139]. Our current RXNet for indole decomposition presents a barrierless reaction channel for the formation of indole from CH_2 and C_6H_5NC, which is presented in Figure 3.17.

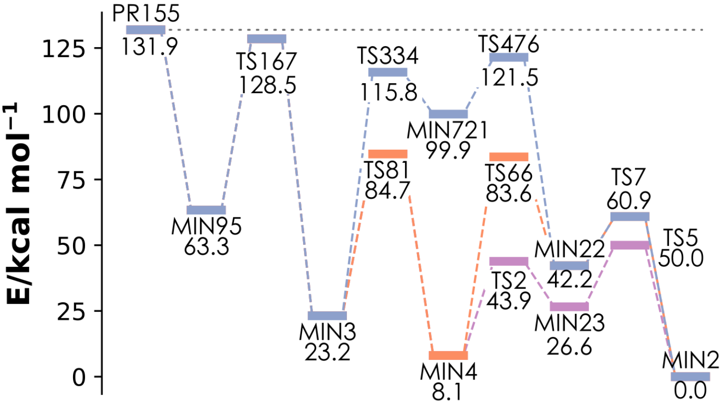


Figure 3.17: Energy profile for a barrierless indole formation mechanism from methylene radical and phenyl isocyanide. Reported energies are potential energies in kcal $\cdot$ mol^{-1}, including zero-point energy. The dashed black line indicates how the whole profile is below the initial fragments (PR155), corresponding to a formally barrierless mechanism.

The barrierless nature of the key fragment association step (PR155 to MIN95) was confirmed through a relaxed scan of the coordinate in the PES of the system, confirming the absence of a transition state. From there, the rest of the mechanism shall proceed smoothly, with the bicyclic intermediate

MIN95 leading to the production of MIN3 (2-methyl-benzonitrile), a species that was also involved in the HCN-forming network (Figure 3.14).

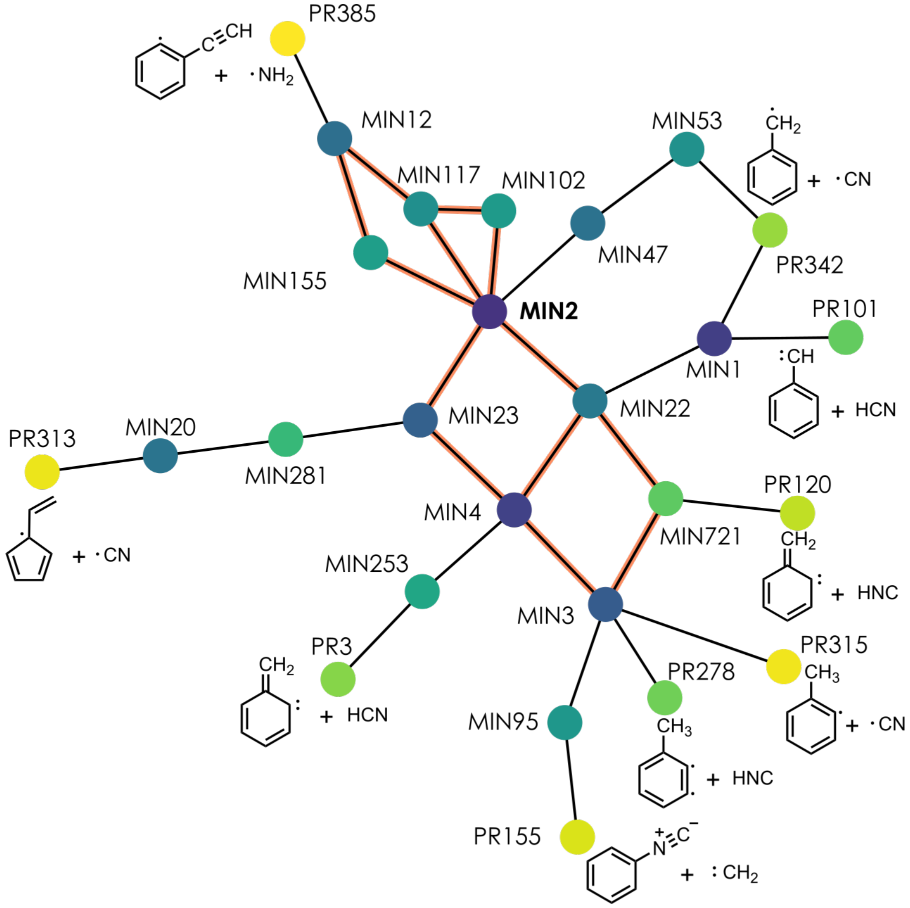


Figure 3.18: Level3 RXNet for indole decomposition, including only species below 150.0 $\mathrm{kcal \cdot mol^{-1}}$. Fragmentation products (PRXXX) are shown as 2D chemical structures. The interconnected "core" of the RXNet, shared between all pathways, is highlighted in orange.

Thus, from this point on, the rest of the mechanism in Figure 3.17 is common with the previously discussed decomposition channels. To highlight this interconnectivity and conclude the overall chemical analysis of the indole decomposition RXNet, the complete Level3 graph including 2D chemical structures for the fragmentation products is depicted in Figure

3.18. This last representation of the reaction network gives two main aspects to discuss. First, it remarks the deep intertwinment of most of the different pathways discussed along the section, except for the more isolated routes producing amino radical in the upper part of the network. For instance, MIN3, a benzonitrile derivative, can give rise to either to cyano radical, hydrogen cyanide or methylene in only one or two steps, as shown in the lower part of the graph. Second, it showcases the chemical nature of the larger fragments formed alongside the small species of interest, which we have mostly neglected until now. These structures correspond to either radicals (for CN and NH_2) or carbenes (for HCN, HNC and NH_2) which are expected to be very reactive, readily undergoing further fragmentation or recombination processes. Whilst this additional cascade of reactions was not part of the current study, as it would have overcomplicated both computations and their analysis treatment, it shall still be acknowledged to properly contextualize our current study. Furthermore, this caveat helps demonstrating how the reaction network graph depiction leads naturally to the expression of systems of ever-growing complexity.

3.4 Conclusions

We have developed a filtering and visualization framework to facilitate the exploration of the complex reaction mechanisms generated by automated reaction space discovery tools, with the aim of making this kind of tools more accessible to the community. While as of now the toolkit is interfaced directly to the AutoMeKin program, its modularity should allow to easily adapt the visualization generation to any other protocol generating reaction networks, as long as the properties included in the dashboards are available. Moreover, we have also taken a step forward on the integration of chemical information on the ioChem-BD database, automating the definition of reaction networks on the platform from also automated mechanistic characterizations.

Besides these visualization capabilities, amk-tools does also allow to simplify reaction networks, segregating the pathways of interest for a given domain from the likely hard-to-treat complete reaction landscape that is usually obtained with fully automated protocols. In this manner, the interpretation of the overall mechanism is streamlined, either through a more traditional point of view based on the extraction of the most relevant energy profiles or through more graph-oriented approaches, depending on the situation. This application of the interchangeability of reactivity representations (Section 2.1) to leverage automated workflows in computational chemistry by making their results more understandable follows, indeed, one of the main objectives of this Thesis.

Finally, the workflow combining automated mechanism discovery with our current strategy for filtering and visualization has been applied to the reaction of decomposition of indole, being able both to reproduce the reaction channels reported in previous DFT studies on the system and to unravel several novel routes, such as the pathways leading to the direct production of hydrogen cyanide and hydrogen isocyanide. All this knowledge on the reactivity of indole shall be relevant in the elucidation of the role of this species in the formation of polluting nitrogen oxides upon coal tar combustion and in the consideration of the presence and possible formation of indole in the interstellar medium.

Chapter IV

Applying the energy span model to complex systems: gTOFfee

4.1 The energy span model

As introduced in Chapter I, the determination of *energies* is, in a general sense, one of the main goals of electronic structure calculations based on Quantum Mechanics. When characterizing a reaction mechanism with computational methods, we propose a set of intermediates and transition states that may arise for our target chemical system and determine how energetically accessible they are. Then, the thermodynamics and kinetics of the process can be understood from the corresponding Gibbs free energies, by locating the lowest-lying products and transition states.

Nevertheless, energies are not straightforward to compare with experimental results to validate the mechanistic proposals obtained from calculations. Experimental reaction performances are not usually characterized through energies, but instead through other magnitudes such as concentrations, conversions, selectivities or rate constants. It is possible to relate these properties to energies through different theoretical frameworks such as Transition State Theory (Section 1.3), getting rate constants from activation energies or considering selectivity differences as differences between the energies of the pathways leading to different products. In other occassions, the other way around can be more useful: transforming the energies obtained by computation to more experiment-friendly parameters. One example of this kind of approach are microkinetic simulations, in which the kinetic behavior of the system is expressed through systems of differential equations characterized by rate constants. The numeric resolution of these systems of equations [140–142] provides the time evolution of the concentrations of the interrelated chemical species, which can then be compared to experimental measurements. This approach has been widely used in the modeling of heterogeneous catalysts [143–147], and it is recently gaining importance for homogeneous systems [148–153].

In the context of catalytic systems, one key observable is the *turnover frequency* (TOF), defined by the IUPAC as the measurement of the efficiency of a given catalyst in terms of the inverse number of molecules reacting per active site at unit time. The TOF (or its inverse, the turnover number or TON) is a very common quantitative assessment of catalytic activity, employed across enzymatic, homogeneous and heterogeneous catalysis [154, 155] to provide general insights on the main question of catalytic design: how good is a given catalyst. In the spirit of bridging experimental and computational approaches, Kozuch and Shaik [38, 156–162] developed the **energy span model** (ESM), which allows to predict the TOF of a given catalytic cycle from its free energy profile in a compact and simple manner (Equation 4.1).

$$TOF = \frac{k_B T}{h} \frac{(1 - e^{\Delta G_r/RT})}{\sum_{ij} e^{(T_i - I_j + \delta_{ij})/RT}} \tag{4.1}$$

In this equation, the ΔG_r term corresponds to the reaction free energy, with the numerator being then related to the thermodynamic driving force of the reaction. This driving force marks the sign of the TOF: for it to be positive, the process must be *exergonic* ($\Delta G_r < 0$).

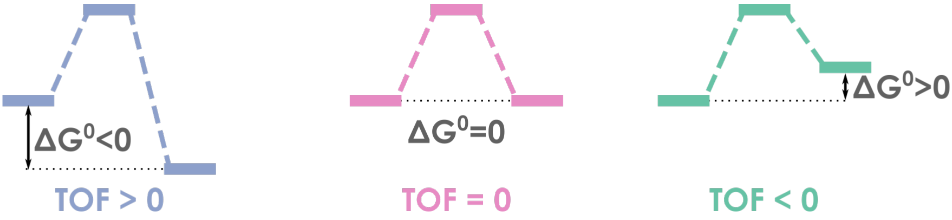


Figure 4.1: Examples of simple reaction energy profiles corresponding to exergonic (left, in blue), isoergonic (center, pink) and endergonic (right, green) processes, indicating the signs of the reaction free energy and of the TOF that will result from Equation 4.1

On the other hand, and as shown in Figure 4.1, endergonic energy profiles would produce a negative TOF, indicating that the catalytic process modeled by the cycle shall go on the opposite direction, which is indeed exergonic. Indeed, "physical" values of the turnover frequency, from the experimental viewpoint, must always be positive, as neither the number of reactive molecules, the number of active sites nor the time can ever be negative. For the limit case where $\Delta G_r = 0$ (isoergonic reaction), the numerator cancels out, with $TOF = 0$ indicating that no turnover occurs for the catalytic system: the catalyst does not work.

The denominator in Equation 4.1 corresponds to the *kinetic resistance* of the process, considering all possible combinations of the energies of intermediates I_j and transition states T_i. The accompanying term δ_{ij} takes into account the relative position of these pairs (between the j-th intermediate and the i-th transition state), taking the value 0 when I_j is *before* T_i ($\mathrm{i} < \mathrm{j}$) and ΔG_{reac} when I_j is *after* T_i ($\mathrm{i} > \mathrm{j}$). This approach allows to properly take into account the cyclic nature of catalytic systems, as an intermediate located after a transition state in the linear representation will indeed need to give rise to that same transition state in subsequent passes along the cycle, as depicted in Figure 4.2. This kind of intermediate/TS pairs involving a TS appearing before the minimum are often neglected when just representing the cycle as a single free energy profile, but can indeed become important to control the reaction.

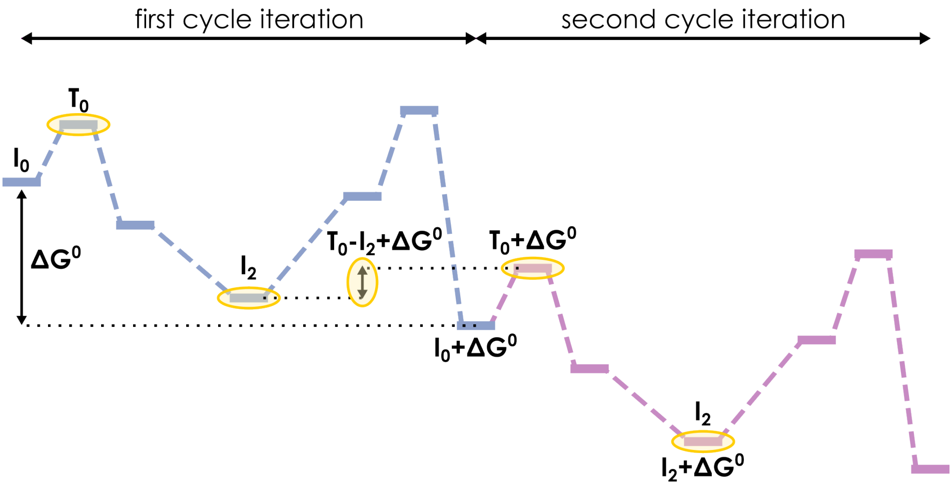


Figure 4.2: Free energy profile for a catalytic cycle, explicitly showing the two first iterations of the process. The highlighted pair (T_0 and I_2) shows the energy difference between the two states that will be introduced in Equation 4.1, incorporating ΔG^0 to account for I_2 being after T_0.

The main advantage of the ESM is that it considers the complete catalytic profile, instead of being limited to a single rate-determining step (RDS) like in more traditional approaches to computational catalysis. Moreover, it does also allow to determine the degree of TOF control (x_{TOF}) of every state in the mechanism, formalizing the "fuzzier" concept of a rate-determining step to the alternative turnover-determining intermediate (TDI) and turnover-determining transition state (TDTS). In this way, it is possible to formally characterize whether a catalytic system is well-defined by a single TDI/TDTS pair that mostly drives the TOF, or whether several relevant states influence the reactivity. In the former case, indeed, Equation 4.1 can be reasonably approximated by the much simpler Equation 4.2, which takes an Eyring-like form and will correspond to modeling the catalytic activity through a single RDS.

$$TOF \approx \frac{k_B T}{h} e^{-\delta E/RT} \tag{4.2}$$

The δE term on the exponent would be the *energy span* naming the model, which is simply the energy difference between the TDI and the TDTS of the cycle, as shown in Figure 4.3.

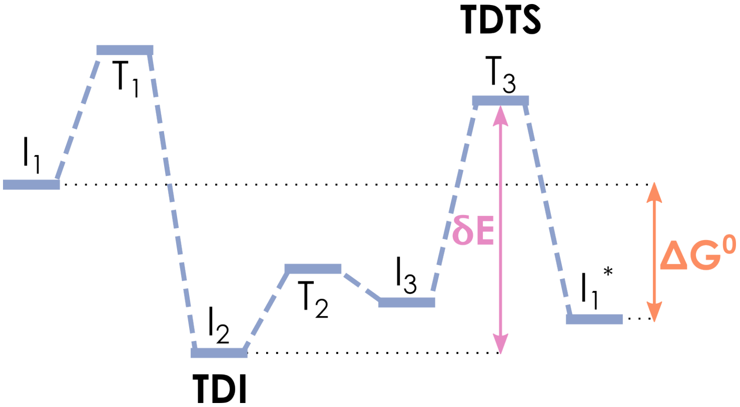


Figure 4.3: Simplified free energy profile with intermediates and transition states labeled according to ESM notation, with I_j intermediates and T_i transition states. Turnover-determining states (TDI and TDTS), the energy span δE and the reaction energy ΔG_r are identified in the profile.

Nonetheless, the utility of Equation 4.2 should not be overestimated: as a proper determination of δE requires knowing the degree of TOF control for the whole profile, it does not actually provide a way to estimate the TOF *a priori*, as the rest of the profile shall still be taken into account to get the TOF control. Moreover, the simplification happening between Equations 4.1 and 4.2 implies a significant loss of detail and predictive power, reducing the utility of the ESM. Therefore, the energy span, as it is, would be mostly useful only for systems that can be summarized in a single TDI/TDTS pair, such as the one in Figure 4.3, to connect the TOF with a simple energetic property. For more complicated systems the complete TOF from Equation 4.1 should be used instead, as TDI and TDTS definitions might not be so clear. For example, coming back to the profile in Figure 4.3, if we had a situation where $I_3 \approx I_2$ (recalling that I_j refers to the energy of the j-th intermediate), it will not be possible to assign any of them as an unique TDI, and the simplified interpretation from Equation 4.2 will fail.

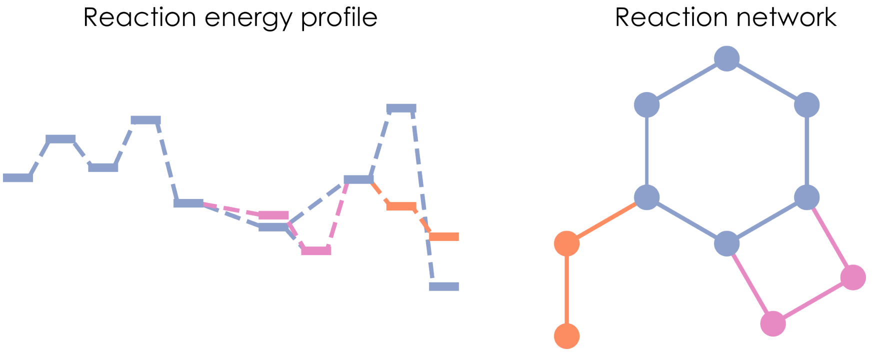


Figure 4.4: Comparison of a reaction mechanism depicted as a free energy profile (left) and as a reaction network (right), as shown in Section 2.1.

Moreover and most importantly, this whole original formulation for the ESM considers only *linear* free energy profiles: while all the states along this profile are taken into account, there is no clear way to handle more complex catalytic systems containing, for example, off-cycle intermediates or side reactions. This follows what we already stated in Chapter II: energy profiles encounter issues to represent entangled reactions, which are better described by reaction networks (Figure 4.4). This supposes a major limitation for the application of the ESM to realistic catalytic systems, whose detailed characterization often produces strongly intertwined mechanisms. Indeed, this issue was acknowledged by the authors of the model and was addressed in its latest developments [163, 164] proposing a much more general formulation based on reaction networks. Nevertheless, since the initial proposal of the graph-based model in 2015, this novel approach had not been yet exploited. In this sense, we developed gTOFfee [165, 166], the very first fully-fledged application of the network-based energy span model allowing to compute the turnover frequency for catalytic systems of arbitrary complexity. Throughout this approach, it is possible to avoid *ad-hoc* simplifications of the input network, keeping all the information of the underlying system.

Before proceeding with the details on the graph-based model and its implementation, we should introduce a couple of additional theoretical considerations about the ESM. First, a relevant feature of the model is the possibility of taking into account the effects of the concentrations of entering reactants and leaving products.

$$\begin{aligned} \Delta G^{\varnothing}(I_j) &= I_j^{\varnothing} = I_j + RT \ln \left[\prod_{h=j}^{N} [R]_h \prod_{h=1}^{j-1} [P]_h \right] \\ \Delta G^{\varnothing}(T_i) &= T_i^{\varnothing} = T_i + RT \ln \left[\prod_{h=i+1}^{N} [R]_h \prod_{h=1}^{i-1} [P]_h \right] \end{aligned} \tag{4.3}$$

Although concentration effects can be critical in the feasibility of a chemical reaction or the overall performance of a catalytic cycle, they are often disregarded in computational studies that directly analyze the free energy profile. As previously mentioned, microkinetic simulations can account for these effects, with their core equations precisely regarding concentration evolutions. However, the setup of this kind of simulations can be quite tricky, especially for more complex systems. The ESM, in contrast, provides a much simpler framework, involving compact algebraic expressions (Equation 4.1) instead of coupled systems of differential equations: this simplicity allows an easier integration in routine workflows. Originally, these effects were introduced in the ESM by a modified form of Equation 4.1 [38], which later on was shown to be equivalent to the assignment of a semi-standard state to Gibbs free energies [164].

This semi-standard state correction considers all catalyst-bearing species to be in a standard state (1.0 M concentration in solution), but corrects reactants and products involved in the catalytic cycle to their actual concentrations.

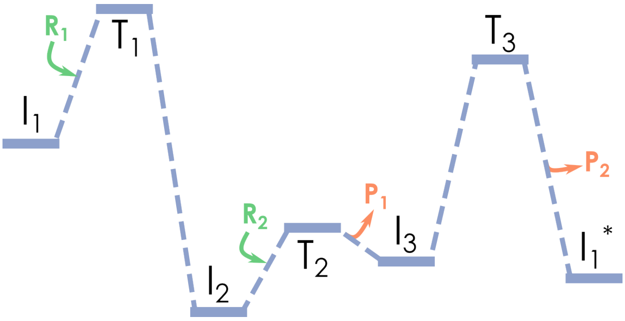


Figure 4.5: Free energy profile labeled according to ESM notation (I_j intermediates and T_i transition states), including entering reactants (in green) and released products (in orange).

This approach might seem unphysical, regarding that the catalyst-containing species would be in very low concentrations and farther from the assumed standard state than these reactants or products. While this concern is relevant and this kind of energies may not be appropriate for general use, the semi-standard approach still provides a convenient and consistent way to introduce these corrections under the framework of the ESM. Therefore, once the energies of intermediates and transition states have been adjusted through Equation 4.3, the resulting TOF values will directly include concentration effects without any further computational overhead.

From Equation 4.3, for a given state I_j or T_i, only the reactants appearing *after* and the products appearing *before* the state are taken into account. Taking the profile in Figure 4.5 as reference, Table 4.1 collects the species' concentrations involved for every state along the sole mechanism in the

I_j	Species	T_i	Species
I_1	R_1, R_2	T_1	R_2
I_2	R_2	T_2	$\varnothing$
I_3	P_1	T_3	P_1
I_1^*	P_1, P_2		

Table 4.1: Entering reactants and leaving products involved in Equation 4.3 for intermediates and transition states from Figure 4.5.

network.

The other relevant theoretical consideration to make is more of a conceptual one: coming back to the initial discussions on how to represent chemical reactivity (Chapter II), the ESM framework highlights yet again the relevance of *energy* as the main descriptor for kinetics and thermodynamics from the computational point of view.

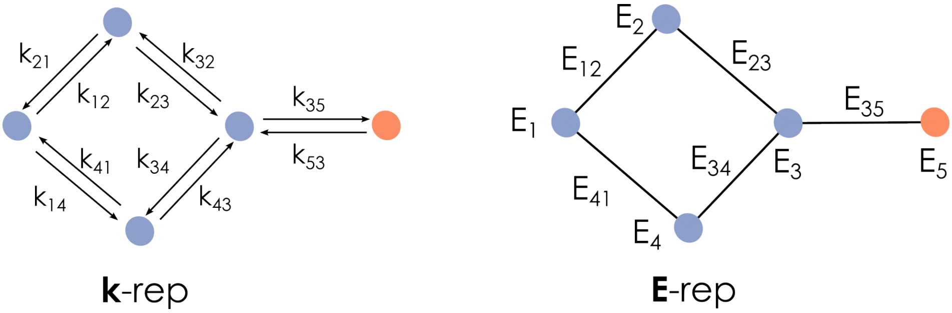


Figure 4.6: Comparison of the directed, rate-constant-based k-representation (left) and the undirected, energy-based E-representation (right) for expressing chemical kinetics in reaction networks.

As stated by Solel, Tarannam and Kozuch [164], we can consider a kind of switch from the **k-**representation, based on rate constants, traditionally used to represent chemical kinetics, and this **E-**representation focused on energies (Figure 4.6). This switch does also suppose a change of paradigm between tackling chemical networks as directed graphs, considering direct and reverse constants for each and every step, or as fully undirected graphs, with traversal information being already encoded in the energies.

4.2 Implementation

A key aspect of the network-based formulation of the energy span model is that it provides a formal definition of a catalytic *reaction mechanism* in terms of Graph Theory. In this context, a mechanism would be a *subgraph* of the complete reaction network including all its nodes (or, in chemical terms, the intermediates of the cycle), but having only a single closed cycle leading to any of the possible products of the reaction. While the full network may have any number of intertwined cycles, valid mechanisms will disregard all but one of these coexisting cyclic pathways.

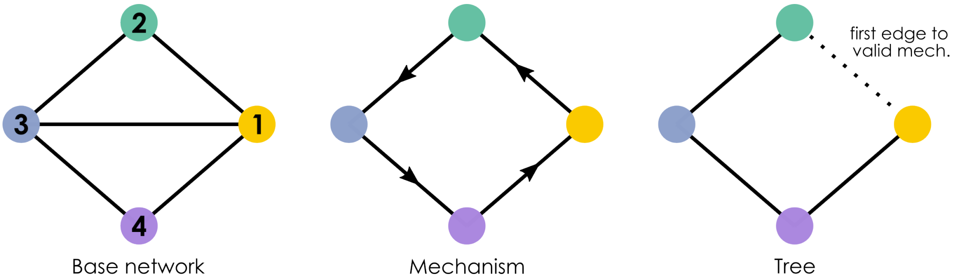


Figure 4.7: Example graph structures for a simple 4-node, 5-edge model network (left) and two kinds of subgraphs: a mechanism (middle) and a spanning tree (right).

Nevertheless, alternative channels are still taken into consideration, as all intermediates are present in every mechanism. In this sense, these channels appear as off-cycle branches that affect the overall reaction, modifying the feasibility of the core cycle of the mechanism. Therefore, from a given reaction network, it is possible to define a set of **n** mechanisms under this paradigm, defining all combinations of possible cycles and branching patterns. Apart from mechanisms, another important type of subgraphs are the *spanning trees*, substructures that still keep all vertices (intermediates) connected, but that do not have any closed cycle. These two kinds of subgraphs and their relationship to the parent reaction network are depicted in Figure 4.7: their specific roles for TOF calculation through the ESM will be detailed in the following paragraphs.

The complete expression for the turnover frequency (equivalent to Equation 4.1) under the graph-based model is shown on Equation 4.4.

$$TOF = \frac{k_B T}{h} \sum_n \frac{\mu_n (1 - e^{\Delta G_r / RT})}{(\sum_k \tau_k) \sum_j (e^{(-I_j + \delta G_{kj})/RT})} = \sum_n (TOF)_n \tag{4.4}$$

In this equation, the TOF for the whole network is expressed as a sum along each of the **n** possible mechanisms arising from the original graph. This system-wide TOF can in fact be regarded as a sum of individual TOF_n terms for each of the **n** individual mechanisms. From the subgraph types introduced in Figure 4.7, Equation 4.4 involves summations across **n** mechanisms μ_n and **k** spanning trees τ_k: thus, these two complete sets of subgraphs must be derived from the input reaction network in order to apply Equation 4.4. Both μ_n and τ_k terms are simple exponentials of the form $e^{(\sum -T_i)/RT)}$, extending the sums to all the *edge* energies T_i appearing along the corresponding mechanism or spanning tree. As the present edges are directly defined upon subgraph definition, the calculation of μ_n and τ_k values from a known subgraph is just a trivial determination of series of exponential terms. Just like in the original formulation, I_j values are just the energies of graph nodes (reaction intermediates), and ΔG_r is the reaction free energy. However, in this case the underlying reaction may be different across different mechanisms, and shall be taken into account as such.

Putting Equations 4.1 and 4.4 side to side, we have that the graph-based variant also has the thermodynamic driving force in the numerator and the kinetic resistance in the denominator, showing a very similar core structure despite the larger complexity of Equation 4.4. To clarify the diverse indices appearing along these expressions, we have that in all cases **i** and **j** identify edges and nodes, respectively, while the more general Equation 4.4 introduces additional indices for subgraphs: mechanisms (**n**) and trees (**k**). It can be demonstrated that Equation 4.4 reduces to Equation 4.1 for simple cycles that can be properly expressed as a *linear* free energy profile, demonstrating the consistency of the two approaches.

The only term that remains to be discussed is δG_{kj}, which as δ_{ij} in Equation 4.1 is related to the relative positions of the intermediates in the catalytic system. While in the original formulation we referred the j-th intermediate to the i-th transition state, in our implementation of the graph framework the reference is done instead to the k-th spanning tree [163].

The assignment of this term is not as straightforward as it was for linear profiles, where the sequential relationship between a given (T_i, I_j) is trivial: when working with graphs, we must deal with the possibility of branching and with the undirected nature of the networks in the E-representation.

$$\delta G_{kj} = \begin{cases} \Delta G_r & \text{if int. after selected edge} \\ 0 & \text{if int. before selected edge} \end{cases} \tag{4.5}$$

The sequentiality rule for a given **node** in a specific **tree** implies seeking whether the j-th node is before or after the first **edge** that would close the k-th tree to one of the previously accepted mechanisms μ_n. Recalling the definitions of both mechanisms and trees, this tree closure implies the addition of a sole edge to the tree to form a single cycle in the resulting graph, as depicted in Figure 4.8, to apply Equation 4.5 and obtain δG_{kj}.

However, neither Equation 4.5 nor Figure 4.8 seem to fully answer the fundamental question: how is this mechanism-defining edge *selected* at all?

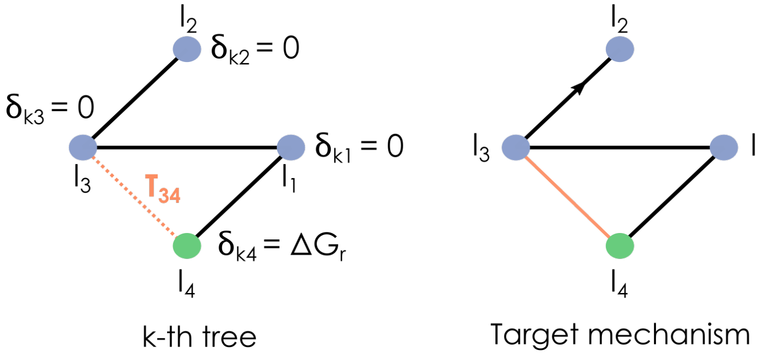


Figure 4.8: Assignment of δG_{kj} for the intermediates in one of the spanning trees derived from the 4-node, 5-edge model network in Figure 4.7. Blue nodes lie *before* the selected edge, while green ones are *after*.

The corresponding rule involves looking for the *first* possible edge that transforms the tree to one of our **n** mechanisms, which requires to somehow traverse the core undirected graph, transforming it to an auxiliary directed graph (or *digraph*) which allows to fetch this information. Without any direction, it will not be possible to know which edges are *before* others, so no first edge could be fetched at all.

This transformation is achieved by specifying what we have named as *closing edges* of the mechanisms (as previously shown in Figure 4.7), that chemically are identified with the reaction steps leading to final catalyst regeneration. The specification of closing edges requires them to end in what would be regarded as the “starting” point of the catalytic cycles, to express the closure relationship. Also, following the specifications of the ESM that were stated before, these edges should always correspond to spontaneous processes, with negative reaction free energy. A reaction network may have several possible closing edges, either leading to different products or just specifying alternate pathways producing the same species. Although the need to define closing edges implies additional information beyond the pure network specification, it is also true that a proper characterization of a catalytic cycle does already involve the definition of a reference state for the catalyst and the step(s) leading to its final regeneration.

Thus, we believe that including this kind of “compass” for guiding traversals across the graph does not really affect the generality of the approach or the advantages of the undirected E-representation. As a final practical note on the topic, in the current implementation of gTOFfee all closing edges are required to begin in non-branched nodes (vertices of degree 2) to properly apply the algorithm. Moreover, the energy of this closing edge should be the reaction energy of the corresponding process, and is used as such to determine the thermodynamics of the process.

All the aforementioned graph “directionalization” is done at the *mechanism* level: indeed, valid mechanisms are required to contain at least one of the closing edges of the network, as depicted in Figure 4.9. It is also possible to find mechanisms having up to *two* closing edges (Fig. 4.9,

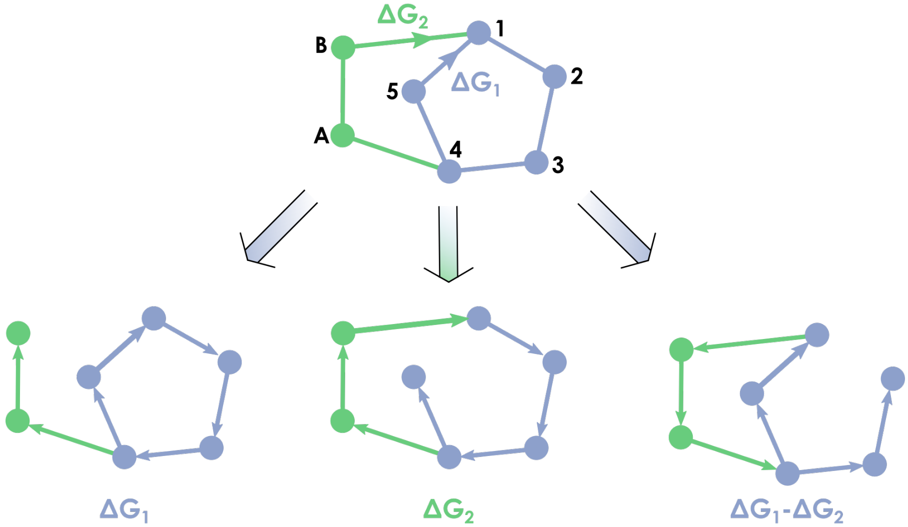


Figure 4.9: Examples of possible directed mechanisms in a toy network containing two possible closing edges, with reaction energies ΔG_1, ΔG_2 and $\Delta G_1 - \Delta G_2$, from left to right.

bottom right). When this situation occurs, only one of them will be the true closure, while the other will be traversed in the opposite direction as another standard edge in the network. Thus, two possible traversal directions arise, from which the one with the most exergonic closing edge will be the one chosen inside gTOFfee. The composite reaction free energy of such processes is just the difference $\Delta G_{reac} = \Delta G_1 - \Delta G_2$, with $\Delta G_1 < \Delta G_2$. Once the traversal direction has been set, all simple paths from the start node to either itself (along the cyclic path) or to any of the possible branch ends are determined. From these simple paths, the lists of edges appearing *before* and *after* any given node are characterized, allowing to store traversal information in the undirected graph skeleton. To handle branches consistently, the point at which they depart from the cycle is used as a reference to assign their location. For example, for the leftmost mechanism in Figure 4.9, node 5 (in blue) will have the nodes A and B in the green branch on its *before* list, as they depart from node 4.

We should recall that this whole discussion on the concept of closing

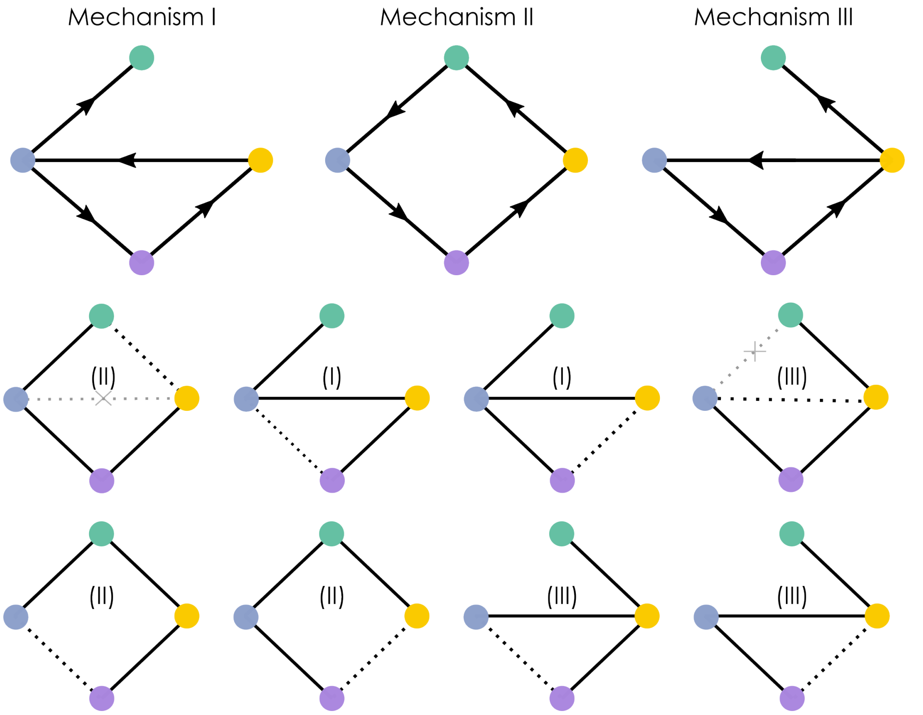


Figure 4.10: Complete set of the three mechanisms (as digraphs) and eight spanning trees arising from a toy 4-node, 5-edge reaction network, as shown in Fig. 4.7. For spanning trees, the selected mechanism-closing edges are shown as dotted lines. Non-used mechanism-closing edges are depicted with crossed dotted lines.

edges and the directionality of mechanisms was motivated by the need to systematically select the first edge transforming a given tree to a valid mechanism, as shown in Figure 4.10. To achieve this, every tree is compared to the full set of mechanisms arising from the network, locating the subset of mechanisms that effectively differ from the tree in a single edge (so the mechanism can be reconstructed by adding this edge to the tree). As every valid subgraph under our paradigm contains all nodes in the network, this can be efficiently done by comparing adjacency matrices. Finally, the edges that convert the tree to each of the mechanisms in this close-able subset are explored, eventually selecting the edge whose end node has *more* edges after it, which can then be thought as being the *first* transforming edge. If several candidates are acceptable (having the same number of edges through this comparison), any of them can be selected. Once this tree-to-mechanism match up is obtained, δG_{ij} terms can be assigned through Equation 4.5 to apply Equation 4.4 and get the TOF values. It is worth noting that as the mechanism-closing edge that we use as reference varies for every tree, the $-I_j + \delta_{ij}$ terms will also vary across trees, and shall then be computed accordingly for each tree. Moreover, all this information is also used to rescale energies according to Equation 4.3, which does also require the relative position of nodes and edges to assign the reactants and products that enter the equation, as shown for linear profiles in Figure 4.5.

Given the name of *energy span model*, the definition of the "energy span" in terms of the graph-based model shows as another important question. In the linear variant the energy span δE was defined as the energy difference between the TDI and the TDTS of the underlying energy profile, which were the states controlling the overall reactivity of the system. The formal definition of these states required to compute the degree of TOF control x_{TOF} of every state along the network, considering intermediates and transition states. The extension of this concept to reaction networks is not a straightforward task, as the expressions used to compute the degree of TOF control [162] are not easy to reformulate in terms of graphs. Furthermore, the concept of reducing the system back to a single TDI/TDTS pair seems quite against our own approach, which aims to handle all the information encoded in the network at once. The most interesting feature of the energy span, in this context, would be to have a descriptor in energy units that we can relate to the overall catalytic activity of the system, having a more natural comparison with other computed magnitudes. Therefore, following this line of thought, we introduced a new magnitude, labelled *effective energy span* δE_{eff}, using Equation 4.2 as a template but considering the exact TOF value resulting from Equation 4.4 at its left side, and then isolating δE_{eff} (Equation 4.6).

$$\delta E_{eff} = -RT \log \left[\frac{h}{k_B T} \cdot TOF_{exact} \right] \tag{4.6}$$

In general terms, what this equation does is a kind of “change of units” in the TOF, transforming its natural frequency units to energies, through an Eyring-like expression. As the $\mathrm{TOF_{exact}}$ value employed in Equation 4.6 takes into account the complete reaction network, δE_{eff} gives information about the overall performance of the catalytic system in the same way as the TOF does. In this sense, we may regard δE_{eff} as a way to measure the *effective activation energy* for a catalytic system. Recalling that Equation 4.4 implies the possibility of decomposing the general TOF for the cycle to the sum of $(TOF)_n$ terms for the individual mechanisms, it is also possible to compute per-mechanism effective activation energies, giving another way to consider the feasibility (and degree of contribution) of every individual mechanism arising for a network. Thus, instead of relying on the location of the most determining intermediate and transition state, we consider all possible branches and steps for both the network-wide and mechanism-wide effective energy spans.

The logarithmic functional form of Equation 4.6 implies that δE_{eff} can only be computed when turnover frequencies are positive and larger than zero or, in other words, when the corresponding catalytic cycle is exergonic. As mentioned in the previous section, only exergonic processes lead to physical TOF values for working catalysts, and therefore it is fully consistent for the effective energy span to only be defined in this case. Another consequence of this logarithmic relationship is that it makes the effective energy span more robust than the turnover frequency, with possible sources of error in the computation of the TOF being translated to much smaller errors in terms of energy. For example, if we had a misestimation of an order of magnitude in a TOF value, the corresponding error for δE_{eff} at room temperature will be around only 1.4 $\mathrm{kcal \cdot mol^{-1}}$. In general, we may frame this effective energy span under the more general context of “apparent activation energies”, following several previous proposals on the matter. For example, a related descriptor based on the degree of rate control (DRC) of intermediates and transition states was recently introduced by Mao and Campbell [167]. There, the authors acknowledge the similarities between this DRC and the degree of TOF control from the ESM, but also the important limitations of the linear-based ESM on the matter, encountering the expected problems in the description of complex systems. This is particularly problematic when putting the energy span defined in Equation 4.2 in the picture, due to the sheer simplification of the catalytic cycle as a TDI/TDTS pair that it implies. Nevertheless, both of these issues have already been tackled by our current proposal: the graph-based model allows a proper treatment of entangled reaction networks and the effective energy span skips the loss of information happening with the "default" energy span.

4.2.1 Computational implementation

After thoroughly discussing the theoretical framework of the graph-based energy span model, some additional details will be given on how the model is coded in our current implementation, gTOFfee [166]. This tool is an open-source code written in Python which is capable of processing any user-defined reaction network to later apply Equation 4.4 and get the corresponding turnover frequency values.

The major points along the workflow of gTOFfee are:

1. Graph representation of reaction networks, using the NetworkX library [69].

2. Generation of all possible unique mechanisms (single-cycle subgraphs).

3. Generation of all possible unique spanning trees (acyclic subgraphs).

4. Assignment of directionality over mechanisms to determine δG_{kj} terms and reactant and product dependencies on edges.

5. Application of Equation 4.4 across all subgraphs, obtaining per-mechanism TOFs.

6. Combination of mechanism-wide TOFs into a system-wide TOF and determination of effective activation energies.

Graph representation

The definition of the input reaction network comprises the very first step of the gTOFfee pipeline. The default format takes two different input files, with one for reaction intermediates (nodes), defining names, energy and connectivity, and other for transition states (edges) specifying the connected nodes, the energy values and the possible species that enter or leave at every given edge. The specification of these reactants and products is employed to include their concentrations, rescaling node and edge energies through Equation 4.3. As the identification of entering (reactants) or leaving (products) entities depends on the direction in which the edges are traversed, the input file assigns them in the direction in which the edge is written, although the true traversal direction will be determined later in the process, and thus the role of a given species as a reactant or product may be reassigned.

This input is then transformed to a NetworkX Graph object, which allows for a very flexible attribute management for both nodes and edges, facilitating the generation of the required subgraphs and their post-processing (e.g., assignment of directionality, energy recalculation...). Apart from the node and edge files and as discussed before, it is necessary to specify all of the *closing edges* leading to catalyst regeneration to indicate the direction of the process.

Beyond the default input, it is also possible to pass graphs in DOT format, as generated in the ioChem-BD platform [82]. While this connection between gTOFfee and ioChem-BD is still under development and does not yet support all intended functionalities (e.g. specification or automatic detection of reactants and products), it provides a simple way to employ the information stored in the database, going through the intended integration strategy hinted in Section 2.5.

Mechanism generation

The formalization of chemical mechanisms as single-cycle, connected subgraphs of the original network is the core concept of the whole extension of the energy span model, as mentioned in the previous sections. From a given reaction network introduced into gTOFfee, the detection of *all* possible mechanisms by manipulation of the input graph is probably the most important task of the program. Given the single-cycle condition, a feasible approach to obtain mechanism candidates from a base catalytic reaction network (which, by definition, must have at least one cycle along its structure) is to remove edges until only an unique cycle remains. To do this, it is necessary to characterize the *minimum cycle basis* of the input network: the set of cycles of minimum length that allow to define the cyclic part (or cycle space) of the graph. Qualitatively, this can be thought as the process of finding the smallest unique cycles which end up forming the target network, as shown in Figure 4.11.

Once the minimum cycle basis N_c is known, the number of edges that need to be removed to form a feasible mechanism is $N_r = N_c - 1$: thus, an exhaustive search of every possible mechanism shall consider all the possible combinations of N_r edges in the base network, raising to a total of $\binom{N_{edges}}{N_r}$ sets of edges whose removal generates a mechanism candidate. For each of these combinations, the resulting subgraph is tested for the three main conditions of a valid mechanism: i) all nodes being still connected, ii) presence of one and only one cycle, and iii) presence of any of the closing edges of the catalytic cycle. At this point, the closing edge of the mechanism marks the reaction happening along its catalytic cycle and the corresponding reaction energy, which controls the numerator term in Equation 4.4. However, given that energies may be transformed to the semi-standard state to include concentration effects, no numeric energy values are stored at this point, but only the target edge or edges that will be used to calculate the reaction energy.

As discussed in the theoretical framework, mechanisms that contain

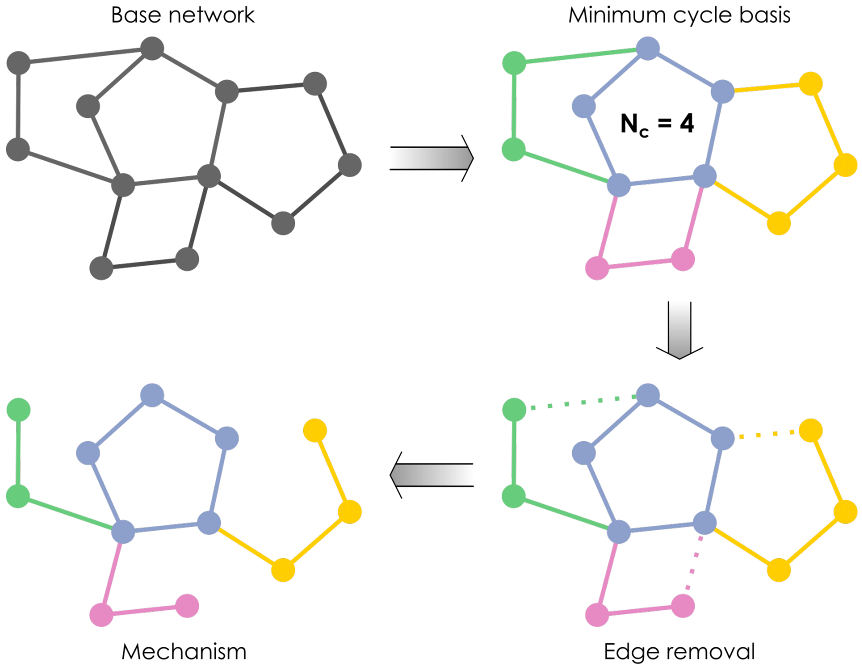


Figure 4.11: Depiction of the mechanism search process on a multi-cycle example network (with $N_c = 4$ cycles), determining the minimum cycle basis and showing one of the possible edge removal proposals leading to an accepted mechanism.

two possible closing edges may appear along the search protocol. In this situation, the energies of the two opposite processes ($G_1 - G_2$ and $G_2 - G_1$) are compared, selecting as closing edge the one leading to a more exergonic reaction to fix the reaction type. Thus, the stored formula for reaction energy will involve the difference between the two edges. Through this strategy, it is possible to handle situations with several intertwined reactions that lead to different products depending on the mechanism.

Tree generation

As from Equation 4.4, it is also necessary to obtain all the spanning trees τ_k coming from a given reaction network. In principle, we may follow a completely analogous procedure to that of mechanisms, considering that in order to get acyclic subgraphs instead of single-cycle ones we should now remove an additional edge: $N_r^{trees} = N_{cycles} = N_r + 1$. Nevertheless, this means that the number of possible tree candidates $\binom{N_{edges}}{N_{cycles}}$ will be remarkably higher than the number of mechanism candidates, increasing the time required to analyze the resulting subgraphs.

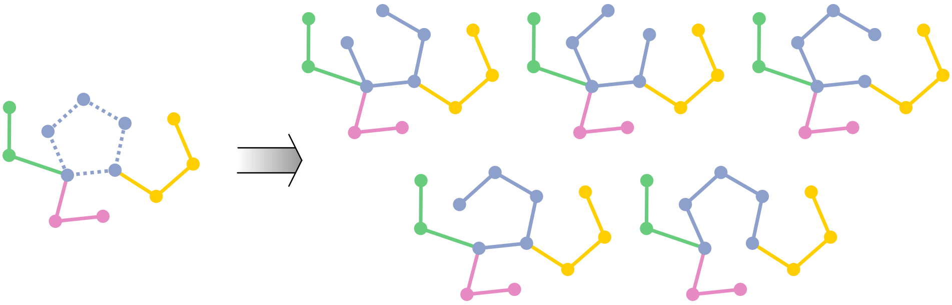


Figure 4.12: Depiction of the tree generation process starting from the final mechanism in Fig. 4.11, leading to five valid spanning trees.

Given that the mechanisms μ_n are found first, it is possible to do the tree search from mechanisms and not from the main network, iteratively removing the edges in their cyclic part as shown in Figure 4.12. This does not only reduce the number of analyzed subgraphs, but also ensures that the final subgraphs will already fulfill the conditions of being connected (as the mechanisms were) and acyclic. The only check that needs to be done through this approach is to confirm the uniqueness of the obtained trees, as it is possible to arrive at the same tree from different mechanisms. This is done by testing whether a newly obtained tree is *isomorphic* with any of the already stored trees, discarding it if this is the case. This isomorphism test, given the properties of the graphs that we are working with, can be done by direct element-wise comparison of adjacency matrices.

To give a sense of the number of subgraph candidates through this protocol, for the network in Figures 4.11 and 4.12 (12 nodes, 15 edges and 4 cycles for the minimum basis), there are $N_r = \binom{15}{3} = 455$ mechanism candidates, accepting 151 of them (33.2 %). If we started from the network, there would be $N_r^{tree} = \binom{15}{4} = 1365$ candidates, which are reduced to 905 by starting from the mechanisms instead. From these, 375 are found to be unique and therefore accepted (41.4%).

Direction assignment

Once all mechanisms and spanning trees have been characterized, it is necessary to find the traversal direction for each mechanism μ_n, based on the closing edges and following the guidelines from the previous section. This graph directionalization serves several purposes: organizing entering and leaving species according to true graph walking sequences, and ordering nodes and edges so as to compute δG_{kj} terms in a per-tree basis through Equation 4.5.

For a given mechanism, the steering process begins by obtaining the tree resulting from the removal of its preassigned closing edge. From this tree, all singly-connected nodes (that is, nodes with degree 1) will correspond to end nodes along the walk, which might be either branch ends or the final point of the cycle. Then, the *simple paths* from the starting node to these end nodes can be computed. At this point, it is possible to get a digraph assigning the directions encoded in the simple paths to every edge in the mechanism (Figure 4.13).

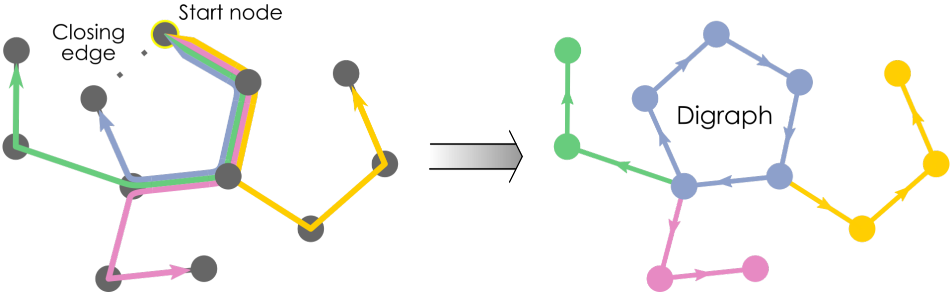


Figure 4.13: Path search and digraph generation from the final mechanism in Fig. 4.11.

Incoming reactants and released products are then assigned for every edge in the digraph, comparing the final traversal direction with the direction in the input file and inverting them if necessary. Finally, iteration along all nodes allows to retrieve information about the edges that are present before and after the target node, and thus the corresponding reactants and products. As mentioned in the theoretical framework, all this traversal information is stored back in the undirected graph core. Then, spanning trees are processed and connected with their corresponding mechanisms by finding the transforming edge which has *more* other edges after itself, as a way to measure the first possible mechanism closure.

TOF calculation

After all relevant subgraphs have been generated and processed, the application of Equation 4.4 to compute the corresponding mechanism-wide and system-wide turnover frequencies is indeed quite straightforward. If concentration effects are requested, Equation 4.3 is applied for every mechanism, taking into account the lists of reactants and products appearing before and after each node and edge. Reaction energies are determined at this point, to be able to include this concentration-based rescaling, fetching the formula assigned to each mechanism upon generation and substituting the energies of the involved edges.

From the denominator of Equation 4.4, labelled D_k, we should note that its second term, which is the one depending on I_j and δG_{kj}, varies across spanning trees (as δG_{kj} does). Relabelling this term as $W_k = \sum_j e^{(-I_j+\delta G_{kj})/RT}$ to highlight the dependency on the spanning tree $\mathbf{k}$, we may rewrite the denominator as:

$$D_k = \left(\sum_k \tau_k\right)\left(\sum_j e^{(-I_j+\delta G_{kj})/RT}\right) = \left(\sum_k \tau_k \cdot W_k\right) \tag{4.7}$$

Through this minimal algebraic manipulation, it is clearer to see how this term is computed inside gTOFfee, requiring the calculation of two exponential terms, τ_k and W_k, for every $\mathbf{k}$ spanning tree. The exponent of the first is the RT-weighted sum of all the edge energies appearing along the subgraph, while the second combines all node energies with the current, tree-dependent set of δG_{kj} values. This D_k term is common to all mechanisms: the specific kinetic resistance of each of the **n** mechanisms is indeed D_k/μ_n, with μ_n being extended to all edges involved in the subgraph. Thus, the per-mechanism TOFs are computed by getting this D_k/μ_n term and the thermodynamic driving force (as in the numerator of Equation 4.4).

TOF post-processing

Although the summation of mechanism-wide TOFs highlighted in Equation 4.4 might seem trivial at first, it actually leaves some room for discussion. Given that a catalytic network may give rise to mechanisms that lead to different products, as we will show in the following sections, the mechanism-wide TOFs should not be naively added up. Instead, it is more valuable to group the mechanisms according to the different products that can occur along the network, only adding up the TOF values inside each of these groups. In this manner, it becomes possible to compare the feasibility of each of the competing routes in the catalytic system, either in terms of the turnover frequency or in terms of its effective energy span. Moreover, these TOFs may be used to provide additional information: for example, in catalytic cycles with two possible products, the quotient of the two TOF values ($\mathrm{TOF_a}/\mathrm{TOF_b}$) was proposed as a *selectivity* measurement [168].

4.3 Applications in homogeneous catalysis: hydroformylation

The initial development of the ESM and most of its applications [156, 159, 169–174] have been devoted to the field of computational homogeneous catalysis, as shown by features like the introduction of concentration effects. Therefore, in order to validate our novel implementation and showcase its capabilities, we selected a well-studied case of study in homogeneous catalysis: olefin hydroformylation.

The production of aldehydes from alkenes and syngas (mixture of CO and H_2) through hydroformylation processes catalyzed by organometallic complexes with metals such as cobalt or rhodium is among the most prevalent applications of homogeneous catalysis in the chemical industry [175–177]. Therefore, there has been a lot of interest on understanding the mechanism of the reaction to consequently tune the catalytic system and streamline its performance, both from the experimental and from the computational points of view. Because of this abundance of information, hydroformylation has become a common benchmark for novel method developments in computational catalysis, which was the reason way we chose it as our main target system.

Provided the existence of very detailed previous computational characterizations of hydroformylation mechanisms, we decided to employ one of these existing studies as our reference: specifically, the mechanism proposed by Rush, Pringle and Harvey [178] reporting Gibbs free energies for propene hydroformylation in presence of the $\mathbf{HCo(CO)}_4$ tetracarbonyl complex (Figure 4.14).

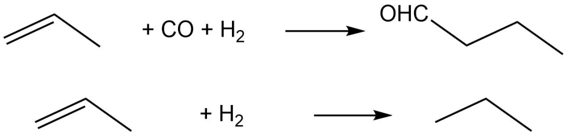


Figure 4.14: Hydroformylation and hydrogenation reactions.

Apart from the mechanistic study, carried out with the B3LYP-D3/6-311G(d,p) functional and basis set, with further refinement of potential energies at the CCSD(T) level, the study by Rush et al. provided a kinetic analysis of the obtained results, showing good agreement with experimental results, which was an excellent point of reference to compare the information collected by gTOFfee.

To be able to introduce concentration effects in our framework, the input Gibbs free energies of the reaction network must be referred to a standard state in solution, with a 1.0 M concentration. However, default values from electronic structure codes are referred to the standard gas phase reference state of 1 atm, requiring to apply a state correction over the energies to modify it. Fortunately, as we are considering the same temperature in both reference states (423 K), this correction does only require a straightforward shifting of free energies, which at our working temperature is 2.98 $\text{kcal} \cdot \text{mol}^{-1}$.

$$G = G^0 + RT \log \frac{P_2}{P_1} = G^0 + RT \log \frac{cRT}{1 \text{ atm}} \tag{4.8}$$

The kinetic simulations carried out in the original study considered an analogous correction, but assuming carbon monoxide and hydrogen to be still in the gas phase: given the framework of our approach, we instead handled these species in the liquid phase too.

	Label	G/kJ mol^{-1}	N_{bodies}	G*/$\text{kcal} \cdot \text{mol}^{-1}$
1+3+H2+CO	1	0.0	4	-23.2
2+3+H2+2CO	2	84.7	5	0.0
4+H2+2CO	3	33.2	4	-15.3
TS5+H2+2CO	3-4	59.3	4	-9.1
6+H2+2CO	4	35.8	4	-14.7
7+H2+CO	5	8.0	3	-24.3
TS8+H2+CO	5-6	65.8	3	-10.5
9+H2+CO	6	48.3	3	-14.7
TS10+H2	6-6B	103.1	2	-4.6
11+H2	6B	21.1	2	-24.1
TS12+CO	6-7	108.6	2	-3.2
13+CO	7	89.9	2	-7.7
TS14+CO	7-8	102.6	2	-4.7
15+CO	8	110.1	2	-2.9
TS16+CO	8-9	123.0	2	0.2
17+CO	9	73.2	2	-11.7
2+18+CO	9X	88.9	3	-5.0
TS19+2CO	4-4B	103.6	3	-1.4
20+2CO	4B	80.0	3	-7.1
TS21+2CO	4B-5X	123.8	3	3.4
2 + 22 + 2CO	5X	x	4	-19.7

Table 4.2: Energies of the hydroformylation reaction from Rush et al. From left to right, species labeled according to the original study, graph tag following our network construction, original free energy at 423 K in kJ mol^{-1} , number of molecular entities in the current step, and corrected 1.0 M-state energy in $\text{kcal} \cdot \text{mol}^{-1}$ (referred to the state 2)

Applying the standard state correction does also account for a certain entropic correction over the energies, as the relative energies of the system will be more or less shifted depending on the number of involved molecular entities. This implies that highly associated states (containing few entities) will be less destabilized than more disassociated states, compensating the overestimation of entropy loss upon association happening when employing usual relative Gibbs free energies. Other alternative corrections have been designed to tackle this specific issue, such as the ones by Martin [179] or Wertz-Ziegler [180–182]: nevertheless, the standard state modification does already handle this effect at some degree. The importance of entropic corrections to match experimental and computed turnover frequencies through the ESM was already highlighted in previous studies by our group [174]. These considerations become even more important when comparing routes with different degrees of association: in the hydroformylation example, the formation of butyraldehyde from propene would be *endergonic* under the original reference state and *exergonic* after corrections, with hydrogenation being exergonic in all cases. Thus, standard state correction shows as an essential asset for a proper comparison of both routes: values extracted from the literature and those employed in this Thesis are shown in Table 4.2.

The focal point of our approach is the leap from the “traditional” representation of a catalytic cycle, in terms of molecular depictions and reaction arrows, to a reaction network in the form of a graph, which can be passed to gTOFfee to apply the ESM and compute the corresponding turnover frequencies. The two representations are shown in Figure 4.15. The network at the right part of the figure contains 13 nodes and 14 edges, defining two distinct cycles. Among the reactants and products participating in the cycle, as it can be shown in the scheme at the left part of the figure, we have hydrogen, carbon monoxide, propene, butyraldehyde and propane, whose concentrations will be taken into account via the semi-standard approach. On this network, gTOFfee finds a total of **12** unique mechanisms and **41** spanning trees. These mechanisms correspond to three unique chemical reactions: alkene hydroformylation (with closing edge **9X - 2**),

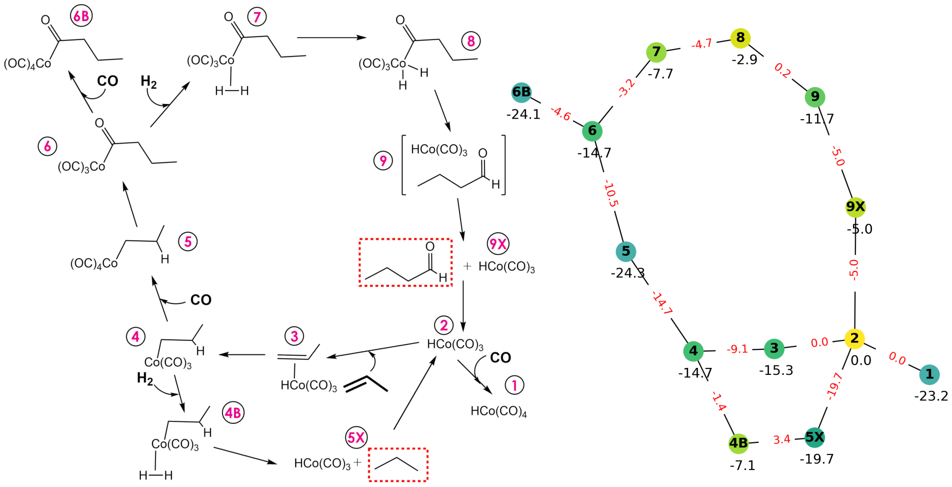


Figure 4.15: Reaction scheme for Co-catalyzed hydroformylation catalytic cycle (adapted from Rush [178]) (left) and full reaction network mapping chemical structures to nodes and edges (right).

alkene hydrogenation (via **5X - 2**) and butyraldehyde decarbonylation (containing both closing edges, with **5X - 2** being the actual closer, to have an exergonic process). These three mechanistic typologies are depicted in Figure 4.16, highlighting the productive catalytic cycle in color.

Looking at the energies in Table 4.2, it shows that the alkane should be the thermodynamic product, with relative barriers for the different pathways not being too different at all. Thus, the insights in reaction selectivity provided by turnover frequencies are valuable to properly analyze the computational data. As mentioned before, in multi-product systems we should group the mechanisms by the product that they lead to and add up the mechanism-wide TOFs accordingly. From there, it is possible to get the effective energy span (Equation 4.6) or to assess the selectivity ratio by the ratio of the turnover frequencies for one or the other product.

As an initial analysis, we will get TOF values for three different situations: i) standard state, without concentration insights, ii) at high reactant concentrations c_{high}: [CO] = 1.5 M, [H_2] = 1.0 M, [Alkene] = 2.0 M, [Aldehyde] = [Alkane] = 0.01 M and iii) at low reactant concentrations c_{low}:

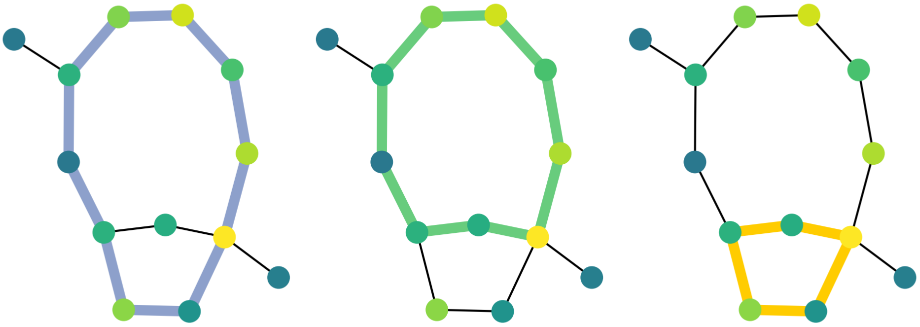


Figure 4.16: Mechanism typologies for Co-catalyzed hydroformylation over the core reaction network. From left to right: butyraldehyde decarbonylation (blue trace), propene hydroformylation (green trace) and propene hydrogenation (yellow trace).

[CO] = 0.2 M, [H_2] = 0.1 M, [Alkene] = 0.5 M, [Aldehyde] = [Alkane] = 0.01 M. The corresponding mechanism-wide effective energy spans, together with the reaction type associated to each mechanism, are collected in Table 4.3. For completeness, the full set of 12 mechanisms is shown in Figure 4.17. In absence of concentration effects (with energies in the standard state), hydroformylation is shown to be kinetically preferred to hydrogenation, with estimated activation energies of 25.2 and 27.9 $\mathrm{kcal \cdot mol^{-1}}$, respectively. If we inspect the individual mechanisms, we find that the only true contributor to aldehyde formation is mechanism M5, while alkane production has relevant contributions from both the decarbonylation route M1 and the hydrogenation route M12, which show quite close TOFs and impact the general reactivity.

When concentrations are introduced, a completely different picture emerges, highlighting the interest of these effects when analyzing complex catalytic systems. At low syngas concentrations (c_{low}), hydroformylation is completely shut down, with a very large effective energy span of almost 40 $\mathrm{kcal \cdot mol^{-1}}$, while alkane production is enhanced, lowering its barrier to 23.4 $\mathrm{kcal \cdot mol^{-1}}$. In contrast, larger quantities of syngas allow hydroformylation to take place, with δE_{eff} values that are very close to the ones in the standard state. This result agrees with known experimental trends and with

Table 4.3: Derived valid mechanisms for the Co-catalyzed alkene hydroformylation. Mechanism index, edge removed from the main graph, G_{reac} and δE_{eff} values, in kcal · mol^{-1}, for the three cases: standard state (c^0), high (c_{high}), and low (c_{low}) concentrations, respectively. Below, conjoined δE_{eff} for aldehyde and alkane-producing routes.

Mech.	Edge	G_r	δE^0_{eff}	δE^{high}_{eff}	δE^{low}_{eff}	Process
M1	2-3	-14.7	28.6	29.5	23.4	Decarbonylation
M2	2-5X	-5.0	48.3	24.9	39.4	Hydroformylation
M3	2-9X	-19.7	33.6	29.6	44.0	Hydrogenation
M4	3-4	-14.7	37.7	38.2	35.7	Decarbonylation
M5	4B-5X	-5.0	25.2	28.9	43.4	Hydroformylation
M6	9X-9	-19.7	33.6	37.3	51.8	Hydrogenation
M7	4-4B	-5.0	30.0	29.8	48.2	Hydroformylation
M8	4-5	-19.7	43.3	42.8	55.1	Hydrogenation
M9	5-6	-19.7	39.1	39.3	48.2	Hydrogenation
M10	6-7	-19.7	31.8	31.6	53.8	Hydrogenation
M11	7-8	-19.7	33.3	33.1	47.6	Hydrogenation
M12	8-9	-19.7	28.4	28.2	42.7	Hydrogenation
Aldehyde			25.2	24.9	39.4	
Alkane			27.9	27.9	23.4	

the kinetic observations by Rush et al., which point at the requirement of large pressures of CO and H_2 to drive the reaction to the desired aldehyde product.

Intrigued by this strong dependence on reactant concentrations and aiming to test the capabilities of our approach we extended our analysis, going from testing individual concentration values, as in Table 4.3, to actually mapping a large set of different initial concentrations. To do this, we considered the quotient $\mathrm{TOF}_{\mathrm{aldehyde}}/\mathrm{TOF}_{\mathrm{alkane}}$ as a selectivity measurement and built a bidimensional map with CO and H_2 concentrations ranging from 0.5 to 5.0 M. The rest of concentrations were fixed, assuming a large concentration of alkene (3.0 M) and some production of both aldehyde and alkane, at 0.25 M. At this point, it shall be noted that under our paradigm we are considering “effective” concentrations for hydrogen and carbon monoxide, despite them being gaseous reactants whose solubility is limited: such large molarities for these species are indeed unphysical.

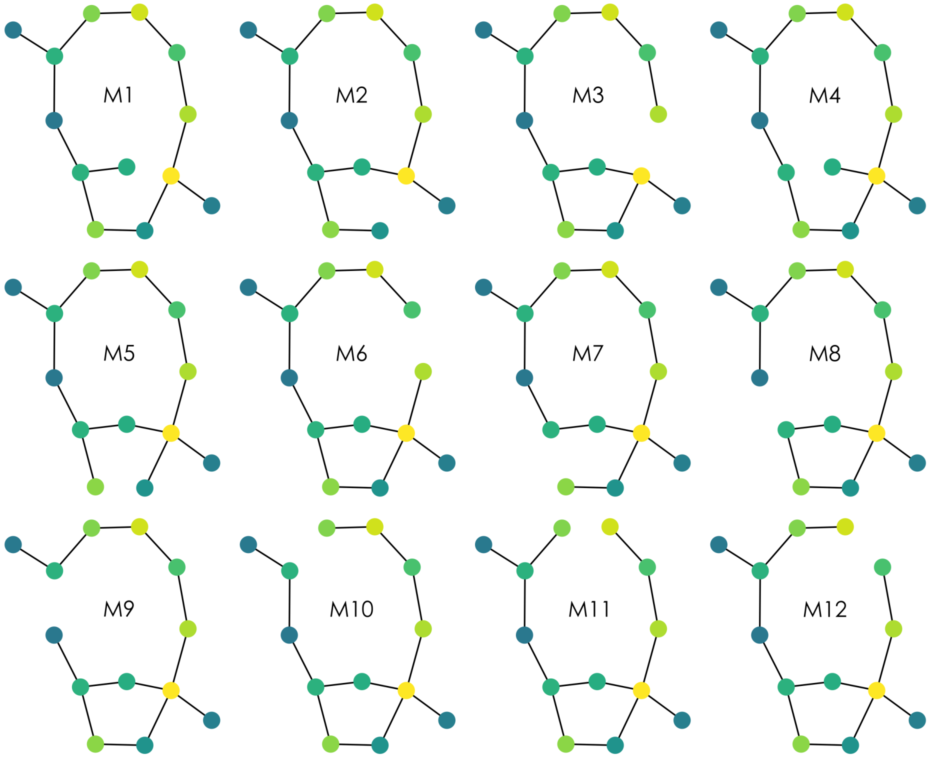


Figure 4.17: Graph representation of the 12 unique valid hydroformylation mechanisms.

However, recalling the foundations of the semi-standard state approach, setting the millimolar concentration values expected for CO and H_2 in solution will imply to have them in concentrations much lower than the catalyst-bearing species, that are set to a standard 1.0 M state. Under this hypothetical catalyst excess, nearly all the dissolved gas will be indeed bonded to the catalyst, resulting in more gas molecules being pulled from the gas phase. In this sense, despite their simplicity, our effective concentrations take this reactant reservoir somehow into account. While a more thorough analysis of the gas phase/liquid phase equilibrium would indeed require more detailed simulations quite beyond the scope of the energy span model, we believe that the current approach still provides relevant insights on

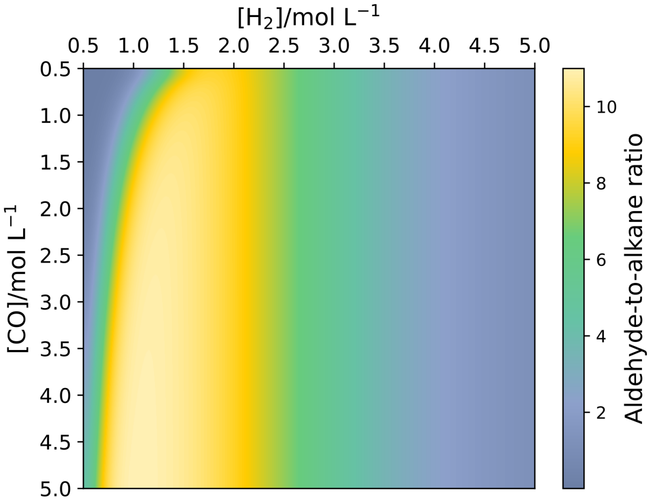


Figure 4.18: Selectivity of the reaction regarding [CO] and [H$_2$], considering effective concentrations in mol · L^{-1}

reactivity trends, as we will showcase in the following paragraphs.

As expected by the preliminary analysis, quite large effective concentrations of both CO and hydrogen are required to properly drive the reaction to the production of butyraldehyde. The selectivity for this product in the upper left corner of the plot, where both concentrations are small, is shown to be really poor, in agreement with the observed experimental performance requiring large pressures of the two gases. Looking inside the hydroformylation-enhancing region (pale yellow zone in Fig. 4.18), we see how the excess of one or the other reactant has dramatically different effects. Increasing the quantity of carbon monoxide (going down in the plot) does not affect the aldehyde selectivity, which remains at its maximum up to 5.0 M. Nevertheless, an excess of H$_2$ has the opposite effect, losing selectivity across the x-axis as the hydrogenation becomes more favored. This can be easily rationalized through chemical intuition: if our goal is to insert carbon monoxide into our scaffold, larger quantities of this species should aid this process. In contrast, larger quantities of H$_2$ will favor its own insertion (hydrogenation), inverting the selectivity trend. Comparing these results with experimental data, we find a very good agreement, as the reaction is indeed carried out with excess of carbon monoxide against hydrogen to maximize selectivity. Experimental and microkinetic-based values for selectivity show values in the range 92 - 98 %, which are in good agreement with the 10:1 ratio predicted in our selectivity map.

We may consider analogous maps for the concentrations of other species (Figure 4.19): nevertheless, the observed effects are much less relevant for the overall reactivity.

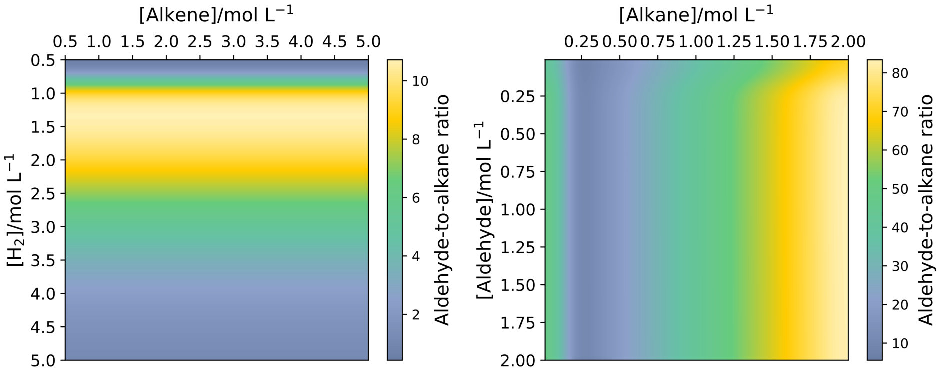


Figure 4.19: Selectivity of the reaction regarding a) [H$_2$] and [Alkene] (ALK) concentrations, b) [Aldehyde] (PR1) and [Propane] (PR2) concentrations. All concentrations in mol $\cdot$ L^{-1}.

Propene concentration (Fig. 4.19, left) has no effect in the selectivity whatsoever, with selectivities being constant across the horizontal axis: this can be rationalized taking into account that the alkene undergoes both hydrogenation and hydroformylation processes, thus having no effect in the relative rate between the two competitive reactions. As for the products (Fig. 4.19, right), it is shown that the reaction is very insensitive to the concentration of butyraldehyde, with decarbonylation routes not contributing much even if a lot of aldehyde has already been generated. In contrast, the excess of propane has a dramatic effect, shutting down hydrogenation and decarbonylation routes. When there is too much alkane in the medium, the selectivity for the aldehyde rises dramatically, showing ratios of up to 80:1 for [Alkane] = 2.0 M. This supposes an additional contribution to the robustness of the catalytic system, as once appropriate quantities of CO and H_2 have been selected, the reaction proceeds effectively and selectively towards the desired aldehyde product. All of these factors contribute to the general adequacy of this homogeneous reaction to be employed at the industrial scale, despite the general preference of the chemical industry for heterogeneous catalysts. Then, the capability of our implementation of the energy span model to extract and highlight these non-obvious features from the reaction mechanism in a simple manner showcases its interest as a tool in computational catalysis.

4.4 Applications in heterogeneous catalysis

Although the energy span model was originally oriented to *homogeneous catalysis*, its core concept (estimating the turnover frequency of a system from its free energy profile) its general enough for it to be appliable to other areas of catalysis. Several examples of application of the ESM to heterogeneous catalysts can be found in the literature, tackling metal surfaces (such as doped Ni [183], doped Fe [184], or Cu [185, 186]), zeolites [187, 188], metal-organic frameworks [189–191] or supported metal catalysts (like Au over ceria [192], Ga over alumina [193], Ga over silica [194] or dispersed Rh [195]). The idea of extending the ESM away from its initial development area (homogeneous catalysis) goes in line with the more general concept of bridging the gap between homogeneous and heterogeneous computational catalysis. While of course both areas will necessarily have their own idiosyncrasies, due to the widely different natures of the catalysts that are modeled, there are common aspects where tools and methods developed for one of these areas can be of interest for the other. A recent example of this kind of efforts is the application of linear scaling relationships (LSRs or *volcano plots*) to homogeneous systems pionereed by the group of C. Corminboeuf [196–199]. LSRs have been a cornerstone in computational heterogeneous catalysis [200–202] for many years, providing a systematic framework for rational catalyst design and tuning from the descriptors generated by DFT calculations, which has not been exploited for homogeneous catalysis until very recently. Indeed, apart from an example of inter-field connection, these molecular volcano plots are directly tied to the ESM, which has been applied [198] to obtain TOFs for catalytic systems and employ them as a descriptor to generate volcanoes. From all this, the ESM arises as a relevant part for this kind of shared toolkit between homogeneous and heterogeneous catalysis that is currently under development.

Previous applications of the ESM in heterogeneous catalysis were limited by the requirement of linearity on the input free energy profiles, whilst situations such as the competition for adsorption sites over surfaces are likely to produce intertwined networks that are not well described by a single energy profile, limiting the accuracy of the resulting TOF just like we discussed in the previous sections. Moreover, and also in agreement to what was discussed for homogeneous systems, further simplifications are made to use the energy span δE as a descriptor, losing an important part of the information collected from calculations. Therefore, the graph-based variant of the energy span model implemented in gTOFfee should provide a more adequate treatment of complex heterogeneous catalytic networks, widening the overall applicability of this approach. Moreover, as discussed at the end of Section 4.1, the effective energy span derived from our implementation provides a more informative measurement of apparent activation energy which circumvents the limitations of Kozuch's energy span.

Nevertheless, we should not overestimate the degree in which the ESM (and gTOFfee) can be directly utilized for heterogenous systems, as despite the improvements over the original model there are still limitations that need to be acknowledged. One of the most evident discrepancies comes from the focus that the ESM puts on concentration effects, assuming species to be in *solution* in order to apply the semi-standard approximation to handle the influence of the quantities of reactants and products. This approach cannot be translated to heterogeneous systems, where the focal aspect is the interaction of the different species with the surface. Because of this, our current implementation is limited to the "bare" turnover frequencies, disregarding the effects of reactants and products. Recent work by Cohen and Vlachos [203, 204] proposes a Modified Energy Span Analysis (MESA) which precisely deals with this limitation, introducing a formal rescaling of free energy profiles based on adsorption rates, in analogy with the concentration-based rescaling through the semi-standard state approximation. While the MESA approach has not yet been introduced into gTOFfee, it shows a promising avenue for the further development of the ESM and its role on connecting homogeneous and heterogeneous catalysis.

Other practical considerations that are required for properly employing gTOFfee on heterogeneous systems involve the way in which the reaction network is defined. Networks in homogeneous catalysis are focused on the different states that the catalyst adopts, following its transformations alongside the chemical processes of interest until the final regeneration of the active species. This kind of state-to-state evolution centered on the catalyst is not as widespread in heterogeneous catalysis, where the specific state of the surface at a given moment might not be well defined. In this situation, reported results include the adsorption energies of the species that occur along the network and the energies of the transition states between adsorbed molecules, but do not define such a clear $A \rightarrow B \rightarrow [...] \rightarrow A$ cycle as there would be in homogeneous catalysis. Therefore, in order to apply the ESM, it can be necessary to "translate" the network, referring the energies of all intermediates and transition states to the all-free-site catalyst plus the reactants, ending up back in the all-free-site catalyst with the corresponding products.

In general, what we aim to cover in this section is mostly a qualitative first approach to employing the graph-based ESM to heterogeneous systems, without aiming to obtain precise TOFs suitable for comparison with experiments or more sophisticated protocols such as microkinetic or Monte Carlo simulations. As hinted before, a better treatment of this kind of systems is one of the possible points of development of gTOFfee, implementing, for example, the MESA approach to provide a true quantitative assessment of turnover frequencies in heterogeneous networks. Nonetheless, we believe that even in this preliminary stage of development, the network analysis provided by gTOFfee makes it valuable as an exploratory tool to have better insights on the chemistry of complex systems in a simple manner.

We will start this part of the discussion with a simple but very paradigmatic example for heterogeneous catalysis: the Langmuir - Hinshelwood (LH) mechanism for a simple $A + B \rightarrow C$ reaction, as discussed by Mao and Campbell [167] in their proposal of a rate-control-based apparent activation energy measurement. The LH mechanism (Equation 4.9) comprises four elementary processes: two adsorption reactions for the two reagents, the combination of the two adsorbed reactants, and the final desorption of the product. In this set of reactions, * represents a free site, while X* corresponds to an adsorbed species and X to a free one (e.g. in the gas phase).

$$\begin{aligned} &1. \qquad A + * \rightleftarrows A* \\ &2. \qquad B + * \rightleftarrows B* \\ &3. \qquad A * + B* \rightleftarrows C* \\ &4. \qquad C* \rightleftarrows C + * \end{aligned} \tag{4.9}$$

To build a suitable reaction network for the mechanism in Equation 4.9, we must consider that any catalytic model for such a system must involve at least *two* active sites, so both A and B can be adsorbed at the same time for the third reaction to occur.

In the context of the "classical" linear ESM, we would need to employ the reaction network depicted in the left part of Figure 4.20, representing a sequential process in which the species B can only be adsorbed *after* the species A. The flaws of such a simplified approach are obvious, as there is no physical reason for the catalyst to adsorb the reactants in this specific order. Moreover, this naive sequential model does also miss the possibility of the two sites being occupied by the same molecule, generating states such as (A*,A*) or (B*,B*). The network on the right side of Figure 4.20

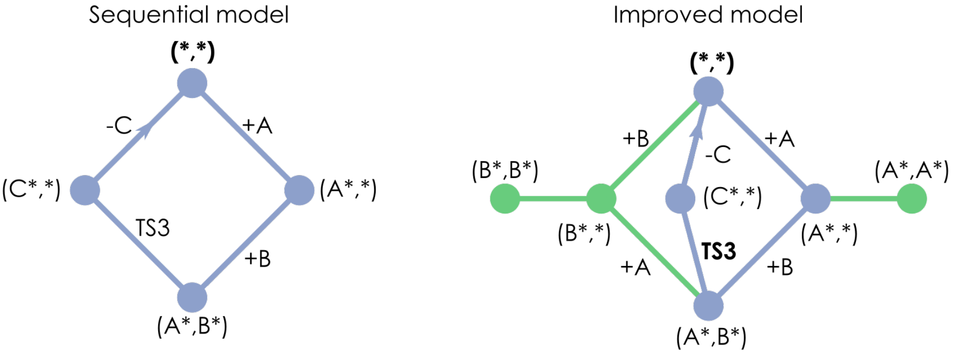


Figure 4.20: Reaction networks for the Langmuir-Hinshelwood mechanism, involving a sequential adsorption scheme (A, then B, left side) and a more realistic proposal (right side) where any molecule can adsorb first and both sites can be occupied by the same species.

does correct these two behaviors, providing a more realistic picture of the Langmuir - Hinshelwood mechanism which cannot be attained with a purely linear free energy profile. This situation clearly exemplifies the importance of switching to the graph-based representation to treat non-trivial reaction mechanisms in the context of the energy span model. In fact, the article by Mao and Campbell presents this specific mechanism as an example of failure of the ESM, where the energy span misses a factor of 2 in the apparent activation energy of the process. In this case study, the transition state TS3 (leading to C* formation) is assumed to be the TDTS of the process and the surface is assumed to be almost covered by A, which implies that A adsorbs more strongly than B (else, the surface will be saturated with B instead). Due to this difference in adsorption energies, the state (A*,*) will be more stable than (B*,*), and consequently (A*,A*) will be lower in energy than (A*,B*) and (B*,B*). Therefore, (A*,A*) should be the TDI of the process: if we recover the original definition of the energy span, we find $\delta E \approx G_{TDTS} - G_{TDI} = G(TS3) - G(A*, A*)$. For non-interacting sites, the adsorption of both molecules of A will be independent, so $G(A*, A*) = 2 * G_{ads}(A)$, leading to $\delta E = G(TS3) - 2 \cdot G_{ads}(A)$, which is consistent with the expression that Mao and Campbell propose in their article in terms of enthalpies ($E_a^{app} = H^0(TS3) - 2H_{ads}(A)$). The factor of **2** accompanying G_{ads} in our expression comes from taking into account the off-cycle intermediate (A*,A*) which is the TDI under the current constraints.

If some of these constraints were to be lifted (i.e., there being other transition states comparable to TS3 or adsorption energies of A and B being comparable), δE will not be a proper descriptor of the activation energy. In contrast, the *effective energy span* computed from the network-wide TOF will be able to capture these effects, despite not having a compact analytical expression, in a similar way to other related descriptors such as Mao and Campbell's apparent activation energy. Overall, this whole discussion on the LH mechanism aims to illustrate conceptually how the reaction-network-based treatment allows the ESM to overcome some of its main issues, while tackling one of the prime examples in the the modeling of heterogeneous catalysts.

To get some additional insights on how gTOFfee can shed light into more complex mechanisms in the context of heterogeneous catalysis, we considered the reaction network for the hydrogenation of CO_2 over a Cu(111) surface, as proposed by Zhao et al. [205]. The corresponding network, shown in Figure 4.21, comprises **28** nodes and **31** edges, with three cycles on its minimal cycle basis. The catalytic system modeled by this network remains relatively simple, but still proposes a different challenge than our previous examples on homogeneous hydroformylation or the Langmuir - Hinshelwood mechanism. Chemically, the underlying reaction is the production of methanol from carbon dioxide ($CO_2 + 3H_2 \rightarrow CH_3OH + H_2O$), with formaldehyde being a possible byproduct ($CO_2 + 2H_2 \rightarrow H_2CO + H_2O$).

Following our previous comments, we needed to adapt the network reported by Zhao et al. to be consistent with the guidelines required by the ESM and gTOFfee. Mainly, every node in the network shall be assigned to a different catalyst state, as done for the LH mechanism, taking into account the occupied and free sites involved in each step. In this case, we did not consider the multi-site occupation of a given species like for the

LH mechanism, keeping the model simple: it should be recalled that our goal on the application of gTOFfee to heterogeneous systems is mostly qualitative. As the reaction energies and barriers in the article are reported in terms of the individual elemental reactions taking place on the surface, we needed to add them along the proposed network in order to get the energies of these catalyst state. To do this, we work under the assumption that already adsorbed species do not affect the co-adsorption of other molecules, so the energy of an A*B* state is just the energy of the bulk plus the adsorption energies of A and B. Finally, it is important to consider that for transition states only zero-point-corrected potential energies (and not Gibbs free energies) are reported, and consequently we will be employing these ZPE-corrected energies for the complete network. Although an accurate TOF determination requires free energies (as it is founded on Transition State Theory), not only our focus is on the qualitative analysis of the resulting graph and not as much on the computation of the TOF, but also the possibility of employing internal energies under certain circumstances was proposed upon the first introduction of the method [206].

After processing the graph in Figure 4.21, we obtain a total of 240 mechanisms and 1117 spanning trees, a vastly larger number than in the hydroformylation example. Recalling Section 4.2.1, the number of subgraphs that can be generated from a certain network depends on its minimum cycle basis N_c: the more individual cycles in this basis, the more sets of edges whose removal must be tested. Although the exact number of valid mechanisms or spanning trees cannot be easily predicted, as it depends on the specific topology of the network, a larger pool of candidates produces a larger number of subgraphs. Here, there were $\binom{31}{3-1} = 465$ mechanisms and $\binom{31}{3} = 4495$ tree candidates, accepting 51.6 % and 24.8 % of them, respectively. From the tree candidates, only 2909 were actually tested, due to being generated from preaccepted mechanisms and not from the core network.

A mechanism-per-mechanism analysis, like we did for the hydroformylation case, would be unfeasible here due to the quantity

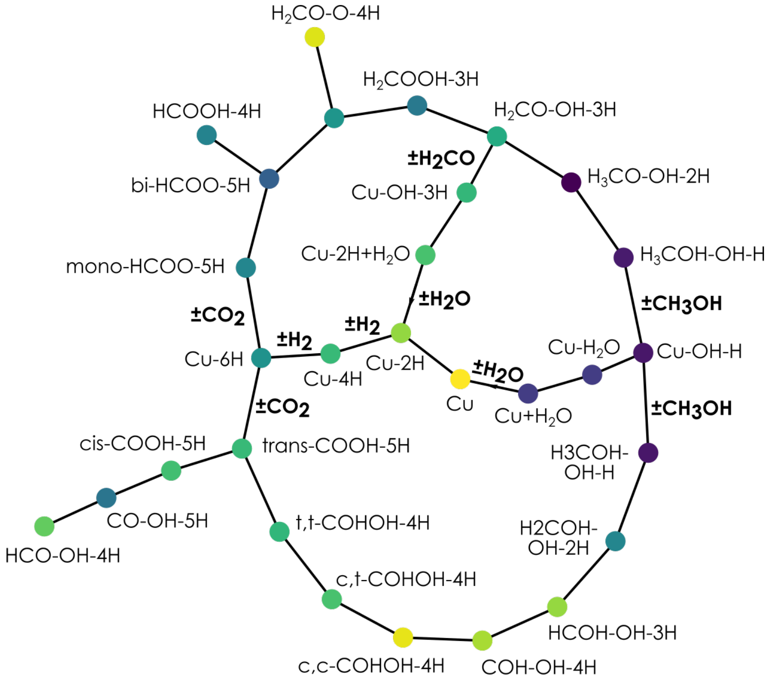


Figure 4.21: Reaction network for the Cu(111) catalyzed reaction between CO_2 and H_2.

of accepted mechanisms. However, these 240 mechanisms correspond to only **six** mechanism typologies (Figure 4.22), involving the production of either formaldehyde or methanol across different intermediates.

Starting from the top left corner of Figure 4.22, we have two routes for producing formaldehyde: R1, starting from CO_2 (blue trace), and R2 (pink) corresponding to a net dehydrogenation of methanol. The other four pathways lead to methanol production: both R3 (green) and R4 (yellow) are formaldehyde hydrogenation routes, while R5 (purple) and R6 (orange) are direct reductions of carbon dioxide. From there, we computed the turnover frequency and effective energy span corresponding to the set of mechanisms leading to each of these six mechanism types, as showcased in Table 4.4.

Looking at the effective energy spans in Table 4.4, we see two clearly preferred routes, which are **R1**, producing formaldehyde from carbon dioxide, and **R3**, producing methanol from pre-existing formaldehyde. While the

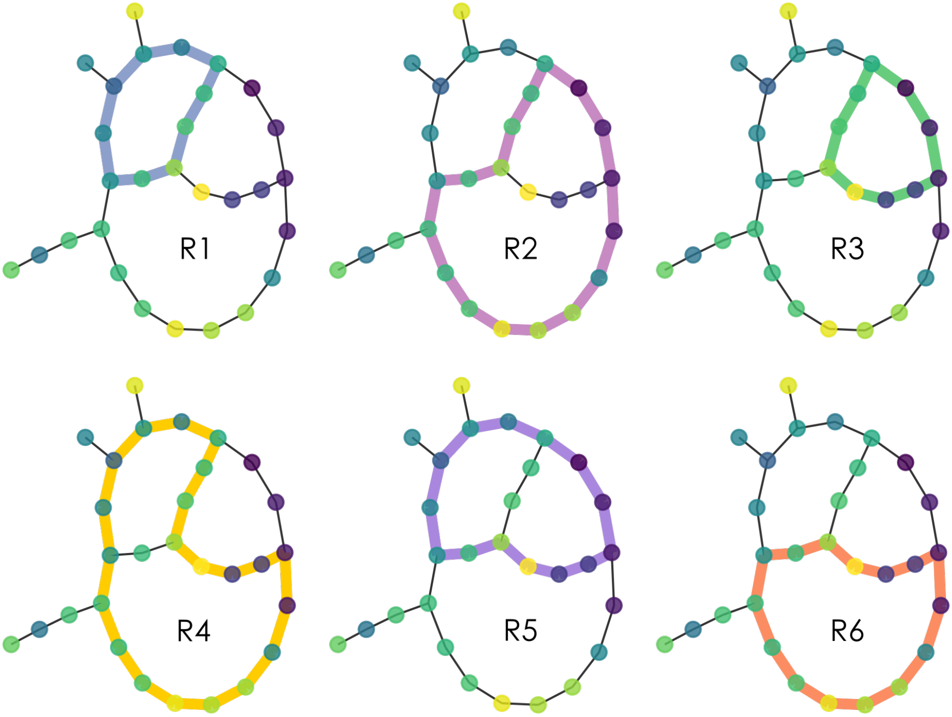


Figure 4.22: Graph depiction of the six unique closed-cycle typologies arising from Cu(111)-catalyzed CO_2 hydrogenation network.

best-performing mechanism for the two routes has the same δE^* value (1.51 eV) for these two pathways, **R3** is slightly more favorable when all mechanisms are added, leading to a δE_{eff} of 1.48 eV.

The reason behind this variation is that in the case of **R3** several individual mechanisms contribute to the overall TOF (and thus to the effective activation energy), while for **R1** only the best-performing mechanism has an impact in the TOF. Taking these results into account, the system is, as expected, confirmed to be adequate for methanol production, as the formaldehyde which is produced through the pathway **R1** will be readily transformed onto methanol through **R3**, with this two-step route being more favorable than the direct pathways R4-6.

To have some more insights on the contribution of the individual mechanisms to the mechanism type they pertain to, we computed histograms

Type	Prod.	N	E_r	δE^*_{eff}	δE_{eff}
R1	H_2CO	75	-0.57	1.51	1.51
R2	H_2CO	20	-0.57	1.99	1.99
R3	CH_3OH	73	-1.04	1.51	1.48
R4	CH_3OH	6	-1.04	2.61	2.58
R5	CH_3OH	27	-1.61	1.73	1.73
R6	CH_3OH	39	-1.61	2.21	2.21

Table 4.4: Summary of mechanism typologies for Cu(111)-catalyzed CO_2 hydrogenation network. All energies in eV. From left to right, typology tag, formed product, number of associated mechanisms, reaction energy, effective energy span of the best mechanism, and effective energy span for the sum of all TOFs.

and smoothed density plots (Figure 4.23) for the per-mechanism effective energy spans in the six groups of mechanisms presented in Figure 4.22 and Table 4.4

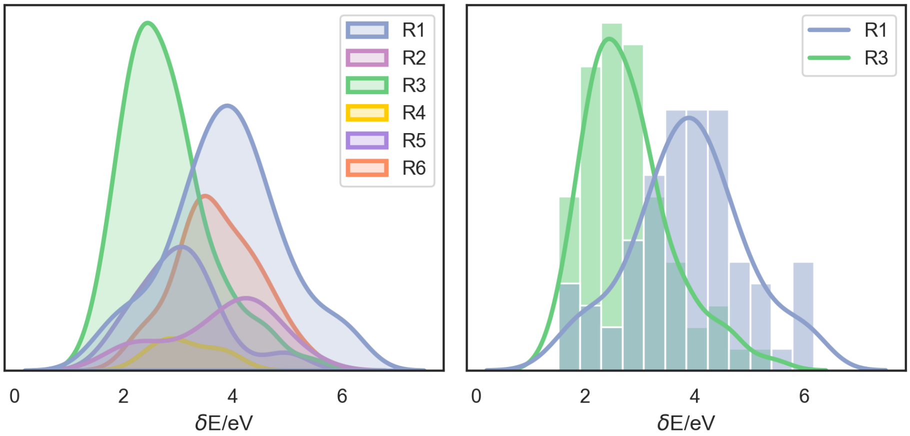


Figure 4.23: Histogram-based analysis of the per-mechanism effective activation energies for the Cu(111)-catalyzed CO_2 hydrogenation network, grouped by the corresponding mechanism typology R1 - R6, with energies in eV. Left, density plots for the six typologies, right, density plots and histograms for the two most favored pathways R1 and R3.

Through the histogram visualization in Figure 4.23 we can get a grasp on how many accessible mechanisms are encoded in every reaction typology. The left plot makes clear how R1 and R3 contain many more subgraphs than the other typologies, whose density plots have lower counts (indeed, R4, with just six mechanisms, is hardly visible in this depiction). We can also see how R2 has the most right-shifted density plot, implying that most of the mechanisms it contains are very disfavored, in contrast with Table 4.4 where its best-performing mechanism had a moderate effective energy span. Comparing the most favored pathways R1 and R3 (right plot), we see how the average R3 mechanism is more accessible (having a lower δE_{eff} value) than the average R1 mechanism, whose curve is remarkably shifted to larger energy spans. From all this, we obtain a more robust confirmation of the advantage of R3-type mechanisms (formaldehyde to methanol) against the other reactive pathways in the network, as most of these mechanisms are reasonably accessible. In general, the idea behind this analysis is to have a more general picture of the reactivity of the system: in the spirit of the rest of the chapter, our main goal is to have a reasonable balance between richness and complexity when analyzing non-trivial reaction networks. Consequently, despite the aforementioned limitations on the treatment of heterogeneous catalysts through gTOFfee, we believe that the qualitative analysis of the CO_2 hydrogenation network shows the strengths of our approach.

4.5 Conclusions

We have developed a computational implementation for the graph-based TOF calculation scheme proposed by Kozuch to extend the ESM to non-linear, complex catalytic cycles. Through this approach, we open the door for a simple, versatile and computationally affordable analysis of realistic chemical systems allowing to go beyond the direct inspection of free energy profiles with minimal effort, including effects that are often neglected in routine DFT studies such as the impact of temperature or concentrations.

While the original concept behind the ESM was mainly the prediction of turnover frequencies, our approach has intended to go beyond these raw values, focusing on the analysis of the subgraphs arising from the reaction network (reaction mechanisms) and on the interest of the effective energy span as a measurement of the apparent activation energy for complex systems. Additionally, our application examples have considered more profound analyses such as mapping selectivities to initial reactant concentrations or an statistical overview (through histograms) of the numerous mechanisms appearing in highly interconnected networks.

Overall, we have shown our approach to be able to reproduce experimental trends in homogeneous catalysis, using hydroformylation as an example, and to have a promising potential for heterogeneous catalysis overcoming several of the limitations associated with the ESM. In general, we believe gTOFfee to be an interesting tool for computational chemists, allowing to obtain chemically relevant information with a minimal setup and under the simple mathematical framework of the energy span model. Going along the major objective of this Thesis, the main strength of gTOFfee comes from the possibility of taking actual advantage from highly detailed reaction networks for catalytic processes instead of trying to reduce them to a single pair of rate- (or turnover-) determining states, as from usual conventions in DFT studies.

Chapter V

Structuring reaction networks as knowledge graphs: OntoRXN

5.1 Semantic organization of chemical knowledge

Semantic approaches permit a very versatile, but still highly ordered, organization of different domains of knowledge, as we introduced in Section 2.3. To express data pertaining to a certain field under this paradigm, it is necessary to define an *ontology* stating the different types of entities appearing in the field of choice and the relationships between them to be able to represent the domain. Nevertheless, ontological approaches to chemistry are still in a quite preliminary phase compared to areas where ontologies are more widely used, like medicine or biology. One of the largest chemical ontologies, ChEBI [207, 208], does fall somewhere in between these two fields, targetting the definition of chemical entities of biological interest. Indeed, the development of ChEBI was prompted by the lack of previous semantic descriptions for chemical data in the contexts of bioinformatics and biomedicine, in contrast with the richness of the ontological description of other aspects of interest for these communities. As of September 2022, this ontology contains almost 60000 annotated entities [209] on small molecules involved in biochemical and metabolic pathways.

Nonetheless, despite the success of ChEBI, the degree of development of other ontologies in Chemistry is moderate: although there is a growing interest on the matter, as showcased in recent reviews [210, 211], no chemical ontology has yet achieved a widespread use nor become a true data organization standard. Because of this, many of the current proposals are still underdeveloped and often quite isolated from the rest of the semantic data ecosystem. However, in spite of these drawbacks, there is a wide variety of relevant efforts tackling plenty of important aspects of Chemistry, which are worth discussing. Sankar and collaborators [212–214] have brought forward a very low-level approach aiming to describe organic reactivity from its most fundamental levels, going from the molecule until its electrons. In sheer contrast, the Named Reactions Ontology (RXNO) [215] comfronts reactivity from a synthesis-oriented approach collecting named transformations to handle general organic reaction schemes. With a somehow similar spirit, the Chemical Methods Ontology (CHMO) [216] considers the description of methods in experimental chemistry, reorganizing laboratory protocols in a semantic manner. Continuing with this heterogeneous and strongly multiscale picture, the concept of Digital Twins [217], aiming to mirror industrial pipelines through detailed computer models, has also been recently applied to laboratory automations [218].

Although cheminformatics and computational chemistry, where the management of data occupies a particularly central position, seem an obvious target for semantic approaches, the impact of ontologies in these has been quite limited. We may highlight CHEMINF, for data-driven cheminformatics [219, 220], or Gainesville Core [221], which collects several basic definitions with the goal of obtaining "a complete description of a typical Computational Chemistry experiment" as reasonably mature developments on the field, but which are still lacking a more general user base. Given that successful ontologies imply a consensus along the community to standardize a certain field, limitations on the adoption of existing ontology proposals eventually end up as limitations on their ultimate development, which requires a collaborative effort. In this line, while a large community wrap-up to define a full ontology including all aspects of e.g. computational chemistry will result on a daunting (and not actually realistic) goal, a more feasible approach involves the development of narrower ontologies tackling specific subdomains and the further connection of the resulting pieces.

Following this idea, we thought that a semantic-based organization would be ideal for the study of reaction mechanisms and reaction networks, formalizing and wrapping up the concepts outlined along the Thesis. Previous ontologies have already considered reaction mechanisms, starting from the point of view of experimental chemical kinetics as in OntoKin [222]. Later on, this description was connected with OntoCompChem [223], a development over Gainesville Core [224], to combine computational studies, kinetics and reactivity. Given that the core of OntoKin is based on experimental kinetics, the kind of mechanisms described by this ontology would be built under the "classical" kinetic framework centered on rate constants. However, we already discussed the convenience of the energy-based **E**-representation to define reaction networks when introducing the energy span model in Chapter IV. Under this paradigm, we can work with simpler undirected graphs that can be directly paired with the energies resulting from computational chemistry protocols, which already encode graph traversal information. Therefore, instead of reutilizing OntoKin, we started building a new ontology for reaction mechanisms founded on the philosophy of the **E**-representation: OntoRXN. The ultimate goal of this project is to produce a standard format combining all the information that is associated to a given reaction network, going from the individual properties extracted from every individual calculation pertaining to the system to the graph structure that embeds the chemical knowledge about the system. To build such an ontology, we considered three main guidelines:

- Application of the **E**-representation, handling reaction networks as fully undirected graphs.
- Conceptual mapping between calculations, species and states along the reaction network.
- Usage of ioChem-BD as the main source of calculation information (stored as CML files) and of the classes and relationships already defined in the OntoCompChem ontology.

The second principle of design is directly related to the way in which we define reaction networks and catalytic cycles, which was already hinted in Chapter III. The states defining nodes or edges along a given network do not usually involve a single molecule or structure, but instead group several species altogether, keeping a balance in the number of atoms across the network. This consistency is essential in order to work with relative energies in chemical systems, so a proper reference state including all involved species can be defined. This reference must be consistent with all the other states in the network, requiring to take into account the points of the network where one (or some) species enter or leave the mechanism.

Finally, the third guideline has a double goal: the integration of OntoRXN in i) the current chemical ontology ecosystem, reutilizing previous definitions for the elements related to individual calculations, and in ii) the ioChem-BD-based data management workflow, converting the non-standard outputs produced by the computational chemistry codes supported by the platform (i.e. Gaussian, ADF, VASP, MOLCAS, ORCA...) to a standardized CML (Chemical Markup Language) format [85–88]. Alternative projects on the semantic organization and publication of computational chemistry results proposed new formats such as CSX [225] to overcome some limitations inherent to CML. However, we believe that the convenience of directly employing the ioChem-BD platform to handle the parsing and storage of information already justifies the direct usage of CML files. Moreover, and as commented on Section 2.5, the possibility of defining reaction networks inside ioChem-BD allows to wrap the complete protocol up, putting the data and its chemical meaning together and facilitating its further ontology-based processing.

As a final note, we shall also consider the connection between our ontology (OntoRXN) and the most similar preexisting proposal combining OntoKin and OntoCompChem [223]. The main discrepancy between the two approaches is, obviously, the switch between rate constants (**k**-representation, OntoKin) and energies (**E**-representation, OntoRXN) as the main kinetic and thermodynamic descriptors of the system. However, both representations can be linked by the transformation of activation free energies to rate constants, through the Eyring equation. We may then design *agents* to traverse the network encoded in a OntoRXN-based knowledge graph (KG), assign the corresponding directions and compute the rate constants for each elementary reaction. Consequently, it would then become possible to link the two ontologies (and the knowledge graphs generated from them) altogether, employing them for their main domains of application: experimental for OntoKin and computational for OntoRXN.

5.2 Ontology development

As discussed on Section 2.3, a central part of ontologies is the class structure which is proposed to organize a certain domain of knowledge. As the *relationships* that power the semantically structured information are defined over these classes, they propagate to the *individuals* defined for a given dataset. Although the actual class structure of OntoRXN will eventually comprise many classes, combining these intrinsic to our ontology to these borrowed from OntoCompChem, Gainesville Core and other related ontologies, we can instead focus on the core classes required to fulfill the aforementioned design guidelines (as depicted in Figure 5.1).

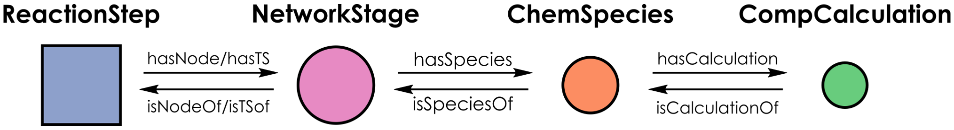


Figure 5.1: Core class structure (topology) for OntoRXN, specifying the four main classes and the properties interlinking them.

These four classes model different levels of the reaction network, obtaining a comprehensive mapping of the individual pieces of information that we outlined in the previous section. The lowest-level class is **CompCalculation**, which represents the results extracted from an individual electronic structure calculation to a CML file in the ioChem-BD platform. The definition of this class borrows directly from the definition of the *GaussianCalculation* class in OntoCompChem, in order to reutilize the properties and relationships that were already defined on this ontology (while also applying some modifications that will be explained later). Then, **ChemSpecies** models the individual chemical entities involved in the network: while CompCalculation refers to a specific, discrete computation result (one output, one CML file), ChemSpecies represents the conceptual molecule for which one or more calculations have been performed. In this way, it is possible to represent situations in which the same chemical species has been the target of different calculations, such as geometry optimizations at different levels of theory, conformational searches, and so on.

Whilst these two first classes describe individual molecules and calculations, the other two model how these entities are connected to build the reaction network. The **NetworkStage** class refers to the set of species (as ChemSpecies entities) that must be considered together to define a proper node or edge (from here on, a *stage*) in the reaction network graph. As introduced in Section 2.1 and at the design principles of the ontology, CRNs for reaction mechanisms should have atom-consistent stages so relative energies across states can be properly computed. To do this inside our ontological framework, we collect sets of ChemSpecies in NetworkStages, which become the basic modeling element for intermediates (nodes) and transition states (edges). Finally, the **ReactionStep** class models elementary reaction steps, which under our graph framework imply an edge and the two nodes it connects. Inside the ontology, this means that a ReactionStep puts together the NetworkStage elements of the intermediates and, possibly, the corresponding transition state associated with the transformation (which might not be present, allowing the representation of barrierless processes). Due to the undirected nature of reaction networks in the E-representation, ReactionStep entities do not contain any kind of properties related to directionality, which would involve concepts such as direct and forward reactions, irreversible and reversible steps... Instead, the assignment of directions becomes a graph traversal problem, just like in the application of the graph-based energy span model (Chapter IV), which will be taken care of, if necessary, by the specific *agents* employing the OntoRXN-based knowledge graphs for one or other application.

Figure 5.2 presents a simple example on how a single elementary reaction step could be represented in terms of our ontology proposal, in order to clarify the aspects that each of the proposed core classes are modeling. The ReactionStep itself (blue square) comprises three NetworkStages (pink circles), corresponding to the two intermediates Int1 and Int2 and the transition state TS1. Both Int2 and TS1 correspond to unique ChemSpecies, which are respectively C and $AB^{\ddagger}$, while Int1 includes two different molecules

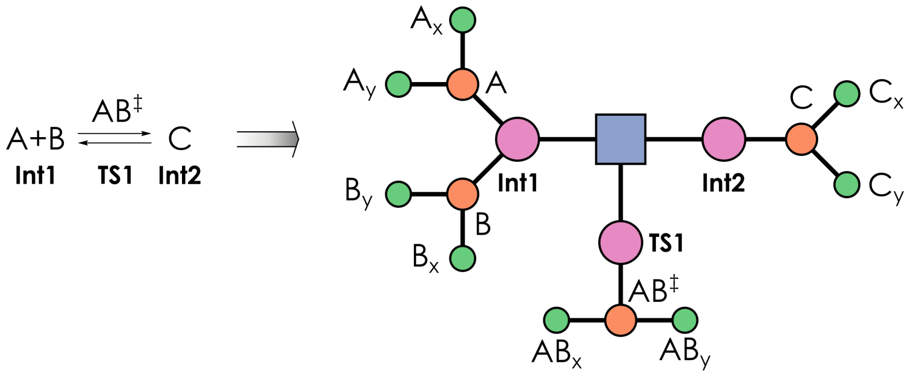


Figure 5.2: Schematic depiction of the complete representation of a elementary reaction step $A + B \rightleftarrows C$ in terms of the class structure of OntoRXN, mapping the entities in the ontology to the labels in the reaction. Regarding calculations, every species is assumed to be computed under two different sets of conditions: $\mathbf{x}$ and $\mathbf{y}$.

(A and B). Finally, every molecule is associated with two calculations ($\mathbf{x}$ and $\mathbf{y}$), which might be, for example, analogous geometry optimizations done at two levels of theory.

Although basing our CompCalculation class on the OntoCompChem ontology is really convenient from the point of view of ontology reutilization and interoperability, a couple of modifications on the borrowed class and its properties were required to properly follow the intended philosophy of OntoRXN.

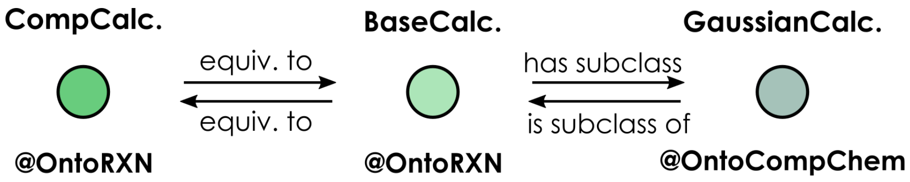


Figure 5.3: Connection between the calculation-defining entities in OntoCompChem and OntoRXN.

The main mismatch is indeed conceptual: the parent class defining a calculation entity in OntoCompChem is labeled *GaussianCalculation*, doing the implicit assumption that only calculations carried out with

Gaussian can be modeled through the ontology. In contrast, our workflow involves the connection of OntoRXN with ioChem-BD, where results are stored in CML format regardless of the package that was employed to run the calculation. Consequently, this program-agnostic approach should be also tackled from the ontology side. To do this, we included a superclass named *BaseCalculation* wrapping up the GaussianCalculation from OntoCompChem (Figure 5.3), aiming to end up defining equivalent classes for all other codes supported by ioChem-BD while still employing the properties and relationships from pre-existing ontologies. At this point, from the four-class structure in Figure 5.1 and its integration with OntoCompChem (Figure 5.3) we should be able to start instantiating knowledge graphs for example reaction networks, initializing an iterative process of development and testing from real computational chemistry data. In this manner, it should be easy to identify the required developmental points for both the ontology and the codes and protocols for the generation of KGs.

It is also worth noting the interest of seeking further connections of OntoRXN with other chemical ontologies, beyond the core integration with OntoCompChem. On the one hand, the *ChemSpecies* class referring to general chemical entities could be easily linked to almost any general ontology including the notion of molecules. This may lead to, for example, assigning PubChem IDs to the species in a chemical network in a semantic framework [226], or, in a more general sense, connecting the molecules participating in the network with chemical data such as their physical properties or reagent sale prices in a complex, multidisciplinary knowledge graph integrating data from several sources. On the other hand, the *ReactionStep* class, modeling elementary reaction steps, might be connected with the kinetic descriptions of OntoKin (Section 5.1) or with the Molecular Process Ontology (MOP) distributed with the more general Named Reaction Ontology (RXNO) for chemical reactions [215]. In this way, the named types of elementary reactions defined in MOP could be linked to the steps present in the OntoRXN-defined reaction network, providing useful metadata for the further processing of the KG. This connection, just like with OntoKin, has the problem of directionality: while the ReactionStep entities do not include this kind of information, labeling any kind of reaction type always assumes a specific direction for the reaction. A solution to this issue could be the definition of *pairs* of reaction types for a given step in the network considering the two possible traversal directions (e.g., oxidation and reduction, fragmentation and association), thus including naming metadata without losing the undirected graph skeleton.

5.3 Generation of knowledge graphs from ioChem-BD

Since recent releases, ioChem-BD includes features for the construction of reaction network graphs from the reaction energy profiles defined in the platform [120], as introduced in Section 2.5. These graphs are an ideal starting point for the generation of knowledge graphs, as they are directly generated from the calculations stored in the platform and are then automatically linked to the individual CML files. In this way, both the raw data and its structure are available in the platform, leaving to the knowledge graph generation workflow only the reorganization of the information to follow the OntoRXN ontology, as depicted in Figure 5.4.

As of now, the workflow has been implemented as a Python interface, named *ontorxn-tools* and published together with the ontology, handling all operations in the pipeline from the retrieval of the information from ioChem-BD to the final instantiation of the knowledge graph.

The very first step of the protocol in Figure 5.4 is, of course, the characterization of the target reaction mechanism: apart from “traditional” manual searches for intermediates and transition states, the protocol can be also applied seamlessly to the automatically discovered networks generated by AutoMeKin (through *amk-tools*, as discussed on Chapter III). In any case, the mechanism predicted from the calculations defines a reaction network

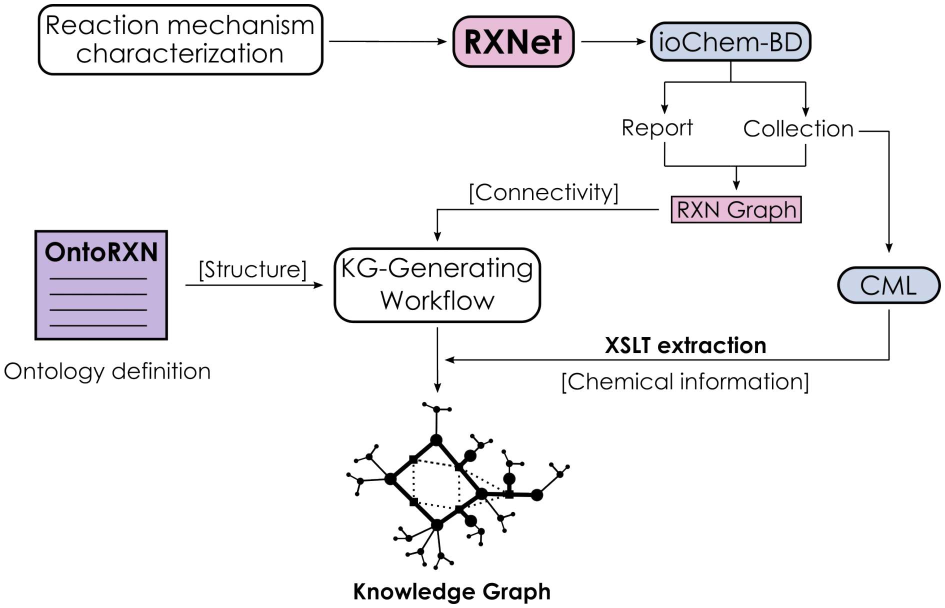


Figure 5.4: Workflow scheme for ioChem-BD/OntoRXN-based knowledge graph generation.

(RXNet, also recalling the nomenclature from Chapter III), which is then pushed to ioChem-BD. This upload stage involves two parts, storing the individual calculations in a *collection* of CML files and the graph structure as a *report* (see Section 2.5 for details on the terminology related to ioChem-BD). By now, report definition requires to transform the RXNet to a set of individual energy profiles encompassing all species and transformations happening in the network, although a more direct definition of reaction network graphs in ioChem-BD is another of our main future goals. Once the set of reaction energy profiles has been defined, it can be transformed back to a network-like representation that can be extracted as a graph, which is used as the actual input for our code. Then, the CML files in the collection, linked to the reaction network through the definitions in the report, can be retrieved through ioChem-BD's REST API.

As shown in Figure 5.5, all retrieved data is, at this point, processed by the KG generation workflow. Roughly, we may say that the CML

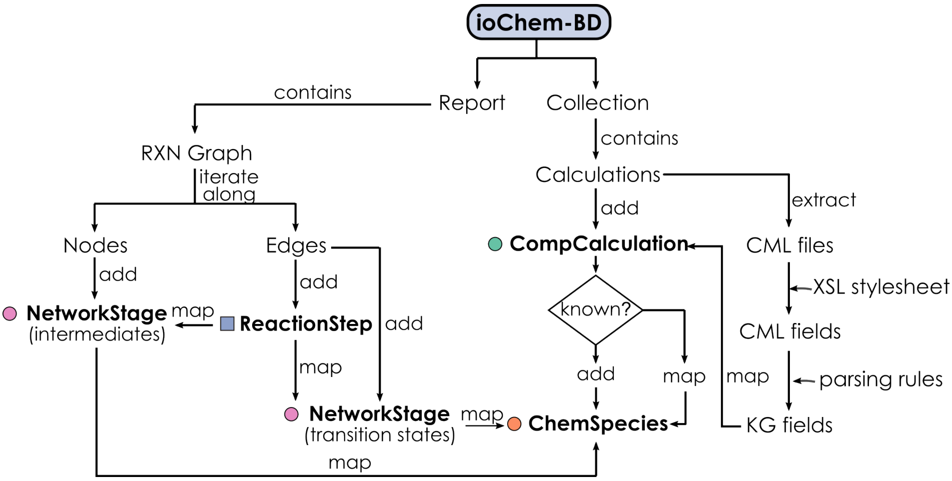


Figure 5.5: Detailed workflow scheme for ioChem-BD/OntoRXN-based knowledge graph generation.

files from the collection will end up defining the CompCalculation and ChemSpecies entities, and the connectivity from the reaction network graph, the NetworkStages and ReactionSteps.

Starting from the right part of the diagram, each CML file is processed by a eXtended Stylesheet Language Transformation (XSLT), mimicking the standard protocol to present and extract information employed across ioChem-BD. This approach involves the generation of specific stylesheets that fetch the information of target fields in the CML file and reorganize them in an alternative format. While in ioChem-BD this format is usually HTML, to directly display the corresponding contents, our code just generates a string of plain text containing key/value pairs mapping the extracted information to a field name. These parsed CML fields are fed to a CompCalculation entity, with the specific assignment depending on the target property. For simple properties (e.g. energy, free energy, InChI string...), there is a set of mapping rules to relate the name of the property in OntoRXN with the CML field name, the data type (numeric, string or vectorial) and its units. This rule collection can be easily modified and extended, simplifying the integration of new properties defined in the ontology. For other properties that require a more complex processing, specific definitions are hard-coded inside the library. For example, in the case of geometry, the Gainesville Core framework introduces individual atom entities containing atomic symbol and XYZ spatial coordinates grouped together in a molecular entity. Then, the block of text containing Cartesian coordinates has to be parsed to properly define the molecular antity and its atoms to follow the pre-established standards.

Once the CompCalculation individual is complete, it is necessary to link it with a ChemSpecies. As these might have a one-to-many mapping with calculations, it will not make sense to naively just generate a new species for every calculation. Instead, the name of the calculation in the report definition is checked and a new ChemSpecies individual is generated and linked to the calculation only if the name has not been yet encountered. Else, the previously defined species is retrieved and mapped. The calculation name was chose as the mapping variable for its simplicity and flexibility, as it allows users to modify the names in the report according to their needs, but it might be possible to use other properties such as the InChI, forcing a more strict mapping.

With all calculations and species being defined, the network graph has to be processed to introduce connectivity, as illustrated in the left part of Figure 5.5. First, the nodes of the graph are iterated through, defining the corresponding reaction intermediates as NetworkStage entities. The same is done for the edges, with each of them defining a new ReactionStep in the KG. These steps are immediately mapped to the NetworkStages of the two nodes it connects, generating and linking a third stage if the edge does also have a explicit transition state structure. Finally, all NetworkStage entities generated during this process are mapped to the corresponding ChemSpecies as from their definitions (or, following from the conventions in the ioChem-BD interface, the *formulas*) in the original report, finalizing the definition of the base knowledge graph. Some relevant relationships in OntoRXN, however, cannot be properly expressed in terms of the OWL language, such as the connection between steps sharing common stages. To handle these situations in a flexible and scalable manner, the code includes a set of SPARQL queries (a querying language for RDF databases that we will introduce in the following section) that infer these additional relationships and add the explicit facts on the knowledge graph to complete it.

5.3.1 Coding details for ontorxn-tools

Just like in Chapters III and IV, apart from the conceptual outline of the workflow, we will provide some additional details on the coding and organization of the overall library. First of all, the package structure (Figure 5.6) involves indeed two different packages: ontorxn-tools itself and *py-iochem*, which wraps several utility functions developed to work with ioChem-BD reports, calculations and its REST API. Both packages, together with the ontology, are freely available on GitLab [227, 228].

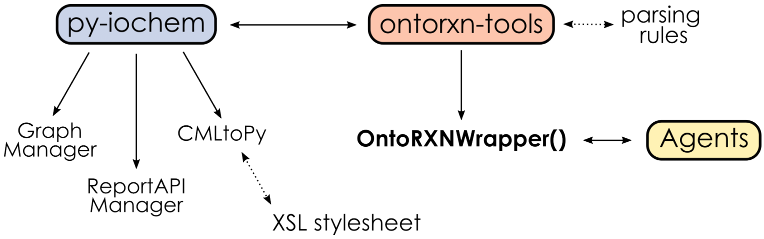


Figure 5.6: Schematic package structure for ontorxn-tools and py-iochem: dotted arrows connect modules with external files required for the main workflow.

Through this structure, it is possible to decouple the ontology-specific aspects from the more general Python interface for ioChem-BD-based assets, facilitating its reutilization, with the ReportAPIManager and GraphManager modules being also employed by amk-tools (Chapter III) or gTOFfee (Chapter IV).

Going on with the operational details, the expected graph input format is, as stated, the DOT graph generated from ioChem-BD reports. If the input network contains disconnected subgraphs, they will be split into individual components, allowing additional flexibility to define several related CRNs (e.g. and as it will be shown in further examples, a network characterized with multiple solvent model parameters) in a single report. Apart from the graph structure, it is necessary to add the ID of the report in the database to properly match the graph itself with the underlying collection.

From there, our code provides the *OntoRXNWrapper* class to simplify the input/output, generation and transformation of the knowledge graphs based on OntoRXN, integrating the ontology management paradigms of two key libraries: owlready2 [229], for general handling of OWL-based entities, and RDFLib [230] for a more general treatment of triple-based RDF graphs. In this way, it is possible to leverage the capabilities of both libraries more easily, with the former being used for general access to the properties and classes of OntoRXN and its knowledge graphs, and the latter for querying the KG as we will detail in the following section.

5.4 Applications: semantic querying of knowledge graphs

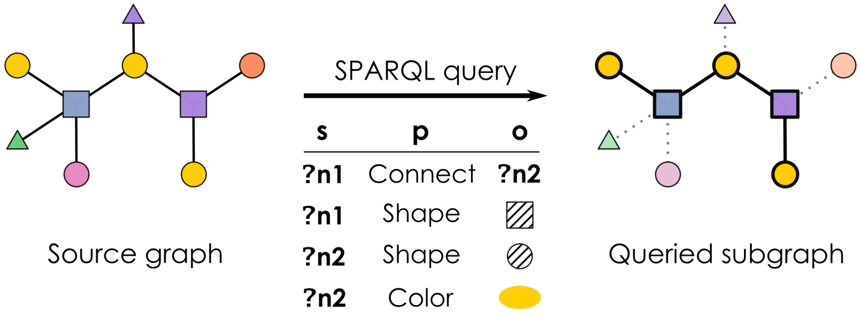
The actual goal of organizing knowledge semantically, from the very first conceptualizations of the Semantic Web, is to permit the formulation of complex questions on the dataset. The knowledge required to answer these questions is inferred from the relationships established not only in the specific target knowledge graph, but also across KGs based in other ontologies that the target KG is linked to. Thus, having a fully ontology-organized domain of knowledge would allow the application of arbitrarily complex queries bringing widely different aspects of the target field together. In the case of Chemistry, these aspects might involve, for example, the energies obtained from a set of computational chemistry calculations, the properties of the solvents employed experimentally to carry out a certain synthesis, the prices of the involved reagents, etc, with these entities being defined in different, but interrelated, chemical ontologies. Thus, it would be possible to ask questions such as *"what would be the most efficient catalyst for a reaction*

X, synthetized in less than Y steps, with source materials not costing more than Z?" and obtain an answer extracted from all the available data in a simple manner. Of course, such a large-scale development is still quite far from reach, given the limited degree of adoption and growth of ontologies on the field. Nonetheless, the semantic querying of smaller KGs founded on narrower ontologies such as OntoRXN arises on its own as an interesting tool for the analysis of complex systems.

Recalling that ontologies and knowledge graphs are founded in the non-relational RDF data model, the way to query this kind of entities is the SPARQL (SPARQL Protocol And RDF Query Language) language [231]. This language is syntactically very similar to the widely known SQL (Structured Query Language), but targetting non-relational databases instead of the relational, table-like databases for which SQL is tailored for. As explained in Section 2.3, the RDF model is based on the assertion of triples, kind of "sentences" connecting a subject with an object through a predicate, which eventually organize the data in flexible and scalable graph structures. SPARQL shares this triple-based organization, but allowing some (or even all) the elements in the triples to be *variable*, generating triple *patterns*. The RDF graph can then be filtered according to the SPARQL patterns, providing subgraphs which only contain the triples that are matched by the query (Figure 5.7)

From Figure 5.7, the proposed example query (below the arrow) considers four triple patterns searching for nodes that are interconnected, with one being a square and the other being a circle, restricted to circles that are also yellow. The application of this query to the RDF source graph in the left isolates the set of five nodes highlighted at the right side.

These subgraphs can be used in two main ways, either adding new assertions to the main graph or simply presenting the extracted information. Queries of the former type are used at the last step of the KG generation process (Figures 5.4 and 5.5) to assert additional relationships in the KG once all entities have been defined, expanding the number of triples in the database. In contrast, the latter permit the enunciation of questions, being



s	p	o
?n1	Connect	**?n2**
?n1	Shape	▨
?n2	Shape	◍
?n2	Color	🟡

Figure 5.7: Graphical depiction of SPARQL pattern matching on a toy RDF graph with nodes having connectivity, color and shape properties. Triple patterns of the query are shown as a table, with the corresponding subject (s), predicate (p) and object (o): entries in bold and preceded with a question mark correspond to variables.

the main tool that the current section is focused on. Several SPARQL endpoints have been already set up to apply this type of queries over semantically-organized chemical databases, such as the Annotated Reactions Database [232] (RHEA) or the Integrated Database of Small Molecules [233] (IDSM).

Coming back to OntoRXN, apart from merely extracting information from the knowledge graphs, it is possible to employ SPARQL querying to design workflows (or, under the naming conventions of the Semantic Web, *agents*) where the data extracted from the knowledge graphs may be directly employed to generate complex plots, prepare structured input for follow-up calculations or, in general, do post-processing on the retrieved information. To demonstrate the utility of KGs and its querying, we will showcase three different examples on computational studies of reaction mechanisms, transforming the corresponding reaction networks to knowledge graphs via the pipeline combining ioChem-BD and ontorxn-tools, and designing specific agents to process the KGs and explore the major points of interest of each specific mechanism. Through this process, apart from highlighting the advantages of SPARQL querying, we will also comment on the most chemically interesting aspects of the involved mechanisms and provide some additional clarifications on knowledge graph structure and generation.

5.4.1 Mapping the knowledge graph: the decomposition of t-butyl peroxyformate

Until now, we have only spoke about knowledge graphs in a theoretical manner, presenting the OntoRXN ontology and the toolkit employed to generate them from concrete datasets. At this point, and before discussing possible SPARQL-based applications, we shall present a specific example of a knowledge graph for a given reaction mechanism, to clarify aspects that might have been overlooked through the general discussion.

The mechanism that we will be using as a reference here is the decomposition of tert-butyl peroxyformate (Figure 5.8), which we recently studied through DFT calculations [234, 235]. This system provided us with a relatively simple mechanism, and consequently a small reaction network, whose corresponding calculations were already available in ioChem-BD. Moreover, together with this base simplicity, the study also offered several points of interest where the KG-based approach could be particularly handy.

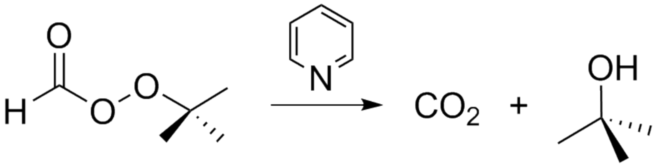


Figure 5.8: Peroxyformate decomposition reaction, producing carbon dioxide and tert-butanol in the presence of pyridine as an organocatalyst.

The interest of the organocatalyzed reaction in Figure 5.8 does not come from its products, which are just a simple alcohol and carbon dioxide, but from its mechanistic intricacies. As observed by R.E. Pincock [236, 237], the decomposition can take place in a large variety of solvents, ranging from very non-polar to highly polar media, with important kinetic differences. The initial mechanistic proposal for this reaction, however, involved a charge-separated zwitterionic complex with a protonated pyridine and a tert-butoxide anion which should not be an accessible structure at all, especially in solvents of low polarity.

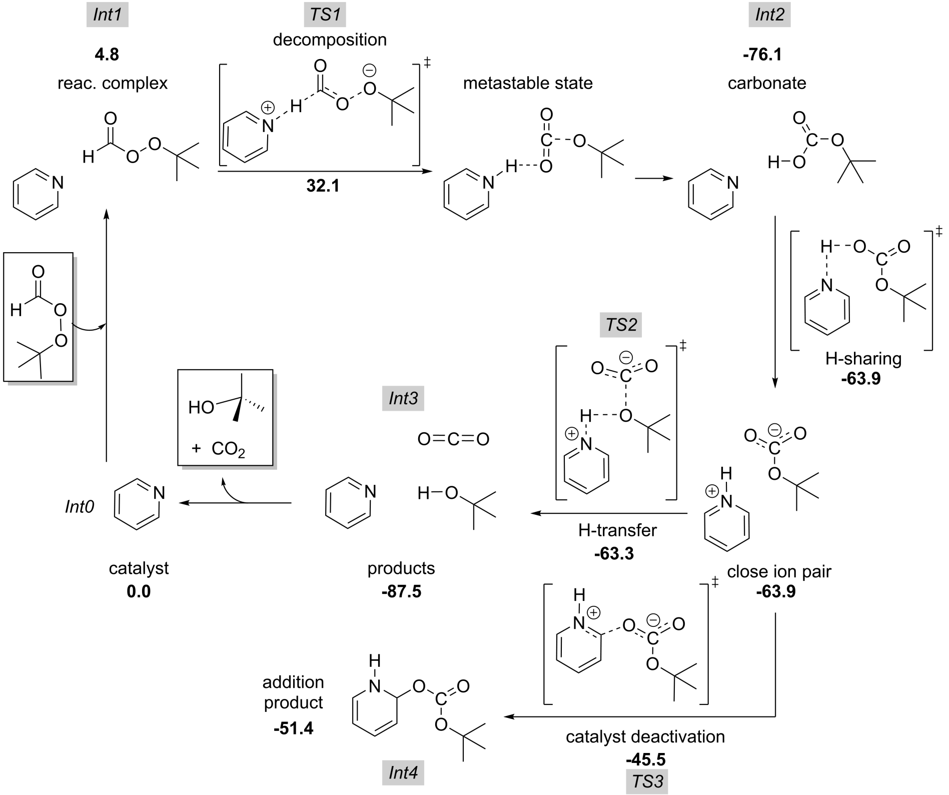


Figure 5.9: DFT-characterized mechanism for peroxyformate decomposition. Gibbs free energies, at a reference state of 1.0 M and 363.15 K, at the ωB97XD/6-311G(d,p) level with SMD parameters for chloroform as a solvent. Entries labeled as *Int* or *TS* correspond to the states employed in the simplified reaction network.

DFT studies[2] (Figure 5.9) allowed us to identify a suitable mechanism in which no charge-separated intermediates had to be formed due to the

[2] Computational details from Ref. [234]. Unrestrained geometry optimizations were carried out using Gaussian09 [238] at the ω-B97XD [239] level of theory, with a 6-311G(d,p) [240, 241] basis set and employing the SMD [242] model for implicit solvation with the default parameters for chloroform. Minima and saddle points were identified by harmonic vibrational frequency analysis, ensuring the presence of 0 or 1 imaginary frequencies, respectively. Dataset collection available in ioChem-BD [235].

ability of the pyridine catalyst to "hold" the proton without it being fully transferred to the base. In this manner, the peroxyformate is isomerized to a previously unreported carbonic acid intermediate, largely more stable than the proposed zwitterion, which then can be readily decomposed by pyridine to release carbon dioxide and tert-butanol.

The rate-determining step is the isomerization of the peroxyformate to the carbonic acid: the originally proposed zwitterion is only a transient metastable state happening along the IRC of the decomposition TS, but not a true minimum on the potential energy surface. Once the carbonic acid is reached, its further decomposition occurs with a quite small barrier ($12.8\ \text{kcal} \cdot \text{mol}^{-1}$). We could also identify a feasible but disfavored route for catalyst deactivation, in which the hemicarbonate anion obtained after deprotonating the acid may attack the pyridine, forming a O - C bond and dearomatizing the pyridinic ring.

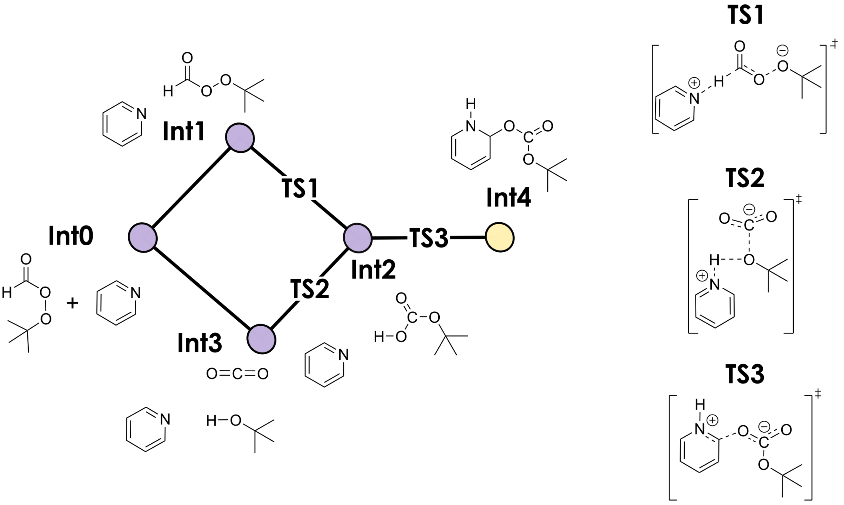


Figure 5.10: Simplified reaction network for peroxyformate decomposition, including the key steps used for characterizing solvent dependence, including chemical structures for intermediates and transition states.

From this mechanism, we considered a simplified reaction network, skipping the details on the proton transfer mechanism from the carbonic acid derivative to model only the main reaction profile. Thus, only the structures with labels shaded in grey in Figure 5.9 are included in the simplified network (Figure 5.10): reference state (separated reactant and catalyst), rate-determining step and product formation. As the main goal of the whole study was the characterization of the solvent dependence of its kinetics, the simplified mechanism was recomputed in a selection of 29 different implicit solvents (see Ref. [234] for details), employing the corresponding SMD parameters.

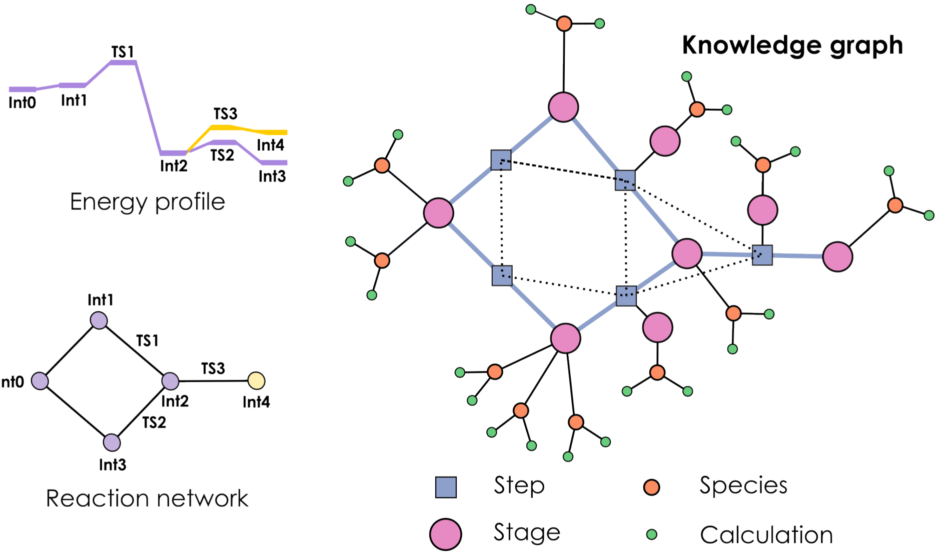


Figure 5.11: Schematic depiction of possible mechanism depictions for peroxyformate decomposition. Above left, reaction energy profile comprising two routes (catalytic decomposition and catalyst deactivation). Below left, basic reaction network structure with nodes as intermediates and edges as transition states. Right, simplified OntoRXN-based knowledge graph, with hierarchical step > stage > species > calculation structure.

From there, the simplified network, considering the results collected for each of the 29 solvents in our dataset, was used to build the knowledge graph depicted in Figure 5.11 together with the corresponding energy profile and network representations. Comparing the knowledge graph representation in Figure 5.11 with its corresponding reaction network, it can be easily seen how the NetworkStage elements (pink circles) somehow preserve the original structure with the rest of the elements highlighting the explicit structure achieved with the semantic approach. For example, it is made clear how some stages group multiple ChemSpecies (orange circles) together and how these species include multiple CompCalculation entities (green circles). For example, we can observe that the **Int3** stage effectively groups together three different species: carbon dioxide, tert-butanol, and pyridine, while **Int1** has only one (the reactive complex). As for the multiplicity of calculation objects, each ChemSpecies is mapped to 29 different calculations, corresponding to the different implicit solvents used for geometry optimizations (although, for simplicity, only two of these 29 CompCalculations are depicted in the figure). Despite the complexity of such a landscape, the structure of OntoRXN allows to naturally express all these relationships in the knowledge graph. Going on with the inspection of the KG, another layer of connectivity can be observed through the interconnection of ReactionStep entities (dotted lines connecting blue squares), demonstrating how the structure of OntoRXN allows to infer additional information from the reaction network. This depiction of the knowledge graph is missing, of course, the actual chemical descriptors that are obtained from the calculations themselves, which would be linked to each of the CompCalculation individuals in the graph. As stated previously, different properties are mapped in different ways, but in general most of them are defined as entities that contain both a numeric value and its units. In the current implementation of OntoRXN, we have focused on a small core property subset to demonstrate our data organization approach, but the goal is to keep expanding the ontology and the properties that are "translated" to the KG from these captured in ioChem-BD.

Once the chemistry of the system has been outlined and the structure of its knowledge graph has been discussed, the main remaining point is the utilization of the KG to demonstrate the advantages of the current ontology-based approach. Using the SPARQL language, it is quite straightforward to write queries to fetch the individual properties pertaining to every CompCalculation entity in the network, find the unique ChemSpecies to which they refer and group them by the NetworkStage entities they participate in. In this way, we can readily generate tables that map every stage along the CRN to any computed property (coordinates, energies, free energies...) or descriptor (e.g species' InChIs). From this basic stage-grouping approach, which is common to most queries of interest for OntoRXN-based graphs, queries can be refined as desired, adding additional filters or search targets depending on the specific request.

The main point of interest of the peroxyformate decomposition reaction was the exploration of the solvent dependence, leading to a KG with a quite large number of CompCalculation entities (290) which are then connected to only 10 ChemSpecies and 8 NetworkStages (5 for intermediates and 3 for transition states). Thus, relevant queries on this system need to target the *solvent* associated to every calculation as a key property to process the knowledge graph. The fundamental question that we could ask for this system could be something such as *"What are the electronic and Gibbs free energies for every stage in the network for every different solvent?"*, whose SPARQL equivalent is presented in Code example 5.1.

```
PREFIX rxn: <http://www.semanticweb.com/OntoRxn#>
PREFIX gc: <http://purl.org/gc/>
SELECT ?stgX ?solvX
  (SUM(?G) as ?Gsum)
  (SUM(?Eel) as ?Eelec)
  (SAMPLE(?nameX) AS ?name)
  (SAMPLE(?epsX) AS ?eps)
WHERE {
  ?stgX rxn:hasSpecies ?spcX .
  ?spcX rxn:hasCalculation ?calcX .
  ?calcX gc:hasResult / rxn:hasElecEnergy / gc:hasValue ?Eel .
  ?calcX gc:hasResult / rxn:hasGibbsFreeEnergy / gc:hasValue ?
      G .
  ?calcX rxn:hasSolvent ?solvX .
  ?calcX gc:hasResult / rxn:hasSolventPolarity / gc:hasValue ?
      epsX .
  ?stgX rxn:hasAnnotation ?nameX
}
GROUP BY ?stgX ?solvX
```

Code example 5.1: SPARQL query to collect electronic energy, Gibbs free energy, solvent and solvent dielectric constant for all stages in the KG.

While a detailed description of the SPARQL language syntax is out of scope, we will use Code example 5.1 to clarify the basic structure and some of key language structures to guide the further discussion. First, the **PREFIX** statement allows to define namespaces for the URIs in the ontology, allowing to use abbreviated property names along the query to make it more readable. Then, the **SELECT** statement determines the fields that should be produced by the query, preceded by question marks to clarify that they are variables. Some of the fields are retrieved directly (in this case, **stgX** and **solvX**), while others are going to be *grouped* according to conditions specified at the end of the query. The keywords **SUM** and **SAMPLE** are *aggregating* functions that specify how the entries to be grouped are managed: the first sums the queried values across the group, while the latter samples a single occurrence of the corresponding values.

The next clause in the query is the **WHERE**, where the relationships used to generate the desired subgraph are defined. This corresponds with the table that was depicted in Figure 5.7, specifying the triple patterns that put conditions on the nodes from the database. In the current example, the entities defining stages, species and calculations are located according to their main dependence relationships (as from Figure 5.1), while the values of calculation properties (electronic energy, Gibbs free energy and solvent polarity) are defined as *results* from the CompCalculation entities. Finally, the **GROUP BY** keyword defines which variables will be used for data grouping, which here are the stage (stgX) and the solvent (solvX). Within this grouping scheme, a total of 232 results are returned, corresponding to the eight unique stages in the reaction network at each of the 29 solvents in the dataset. Without grouping, we will obtain a longer table (319 records) where for a certain stage/solvent combination we will have as many entries as species participating in the stage, which would need further processing to obtain stage-wide properties such as energies.

Through the results of the SPARQL query in Code example 5.1, we may easily obtain the collection of the 29 free energy profiles describing the peroxyformate decomposition mechanism for the solvent set (Figure 5.12)

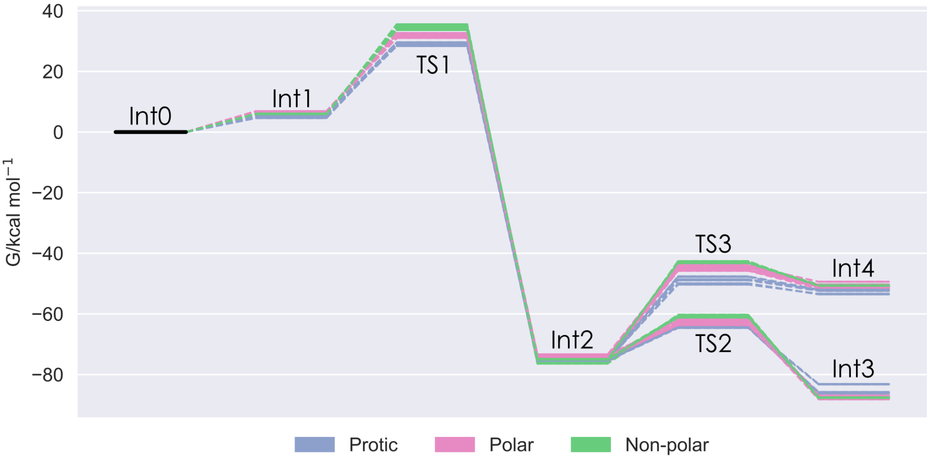


Figure 5.12: DFT-characterized free energy profiles for peroxyformate decomposition. Gibbs free energies, in kcal · mol^{-1}, at a reference state of 1.0 M and 363.15 K, at the WB97XD/6-311G(d,p) level with the SMD solvent model for the set of 29 solvents presented in Ref. [234].

To properly analyze solvent dependence, we should extract the activation energy from the set of profiles in Figure 5.12, knowing that it corresponds to the rate-determining step (peroxyformate isomerization, from Int0 to TS1), and compare it with a solvent parameter such as its dielectric constant. The results of this analysis, considering both potential and Gibbs free energies, are collected in Figure 5.13.

We can see the solvent dependence clearly reflected on the computed barriers, but instead of a smooth decreasing trend, we observe a more profound relationship with three very distinct groups that match perfectly with the expected solvent natures: non-polar (with largest barriers), polar and protic (with remarkably lower activation energies). In fact, the colored subsets in Figures 5.12 and 5.13 come from data-driven groupings through a K-Means method over the $\Delta E/\log\varepsilon$ data, which match perfectly with the expected qualitative non-polar, polar and protic labels. The KG-based workflow allows to produce these plots in a very easy and automated manner which could be seamlessly scaled to larger sets of solvents (or other

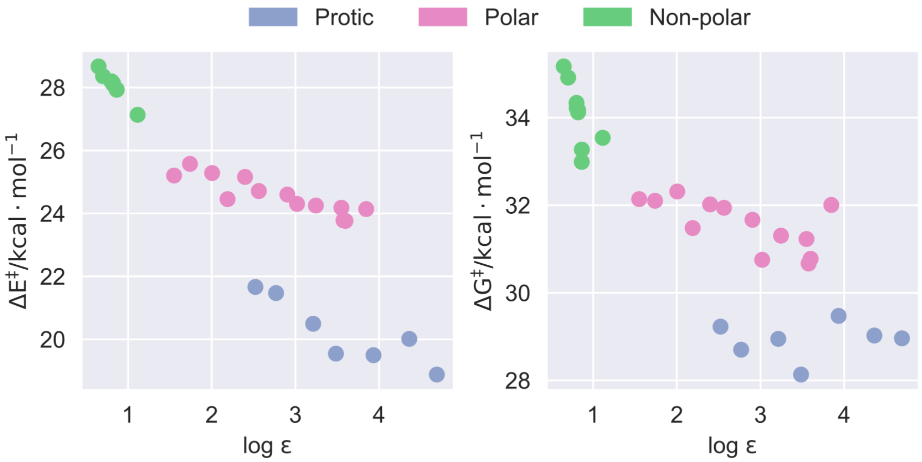


Figure 5.13: Barrier heights (electronic and free energy), in kcal · mol^{-1}, for peroxyformate decomposition against the logarithm of the dielectric constant of the solvents in the dataset [234]. Colors correspond to solvent types (non-polar, polar and protic).

parameters), providing a powerful solution for complex mechanistic studies.

Apart from generating reaction energy profiles, the KG can also be used to recover the original reaction network that it encodes. Although this might seem an unnecessary circular detour, producing a knowledge graph from a reaction network and using the KG to regenerate the same reaction network, it can be valuable in multiple situations. First of all, recovering the CRN from the KG allows to effectively employ OntoRXN-based knowledge graphs as a standard format for reaction networks, so the KG wrapping up all the information on the system can be shared and then transformed to a more readable reaction network. Moreover, it provides an immediate interface of KGs with any workflow or program employing reaction networks as its input, such as gTOFfee (Chapter IV). Finally, the back-transformation process also makes possible to generate reaction networks containing any subset of the information in the knowledge graph, acting as a structured CRN generator.

To carry out this transformation, we need a SPARQL query to retrieve the connectivity of the NetworkStage entities for *nodes*, whose structure conforms the reaction network. Additionally, we should also map these edges having an associated transition state with the corresponding NetworkStage, as in the query proposal shown in Code example 5.2.

```
PREFIX rxn: <http://www.semanticweb.com/OntoRxn#>
SELECT ?stepX (GROUP_CONCAT(?stgX) as ?stgL)
  (SAMPLE(?stgY) AS ?stgTS)
WHERE {
?stepX rxn:hasNode ?stgX .
OPTIONAL {?stepX rxn:hasTS ?stgY}
}
GROUP BY ?stepX
```

Code example 5.2: SPARQL query to extract basic network connectivity from the knowledge graph, locating the two nodes connected by each step and, if present, the corresponding transition state.

A couple of additional keywords which did not appear in Code example 5.1 are present here. On the one hand, **GROUP_CONCAT** is, like **SUM** or **SAMPLE**, an aggregating function that in this case concatenates a set of strings with a given separator. On the other, the **OPTIONAL** keyword allows the specification of statements that can or can not be found in the RDF database, returning an empty value in the latter case. In the absence of **OPTIONAL**, if a requested variable value is missing, the whole record involving the missing value will be omitted from the search, which in this case would imply to lose the steps not having a transition state.

As the connectivity query ultimately refers to the stages of the KG, it can be combined with any kind of property-finding query: for example, the one in Code example 5.1 to add the energy of each stage. Coming back to the peroxyformate KG, this approach will allow to obtain individual reaction network depictions for every solvent in the dataset (Figure 5.14) complementing the profile-based view. In general, the connectivity query makes use of the two separate properties that link ReactionSteps and NetworkStages: *hasNode*, which will always appear twice for every step, finding two connected nodes, and *hasTS*, which may or may not be defined depending on the existence of a TS for a certain edge. The results of the

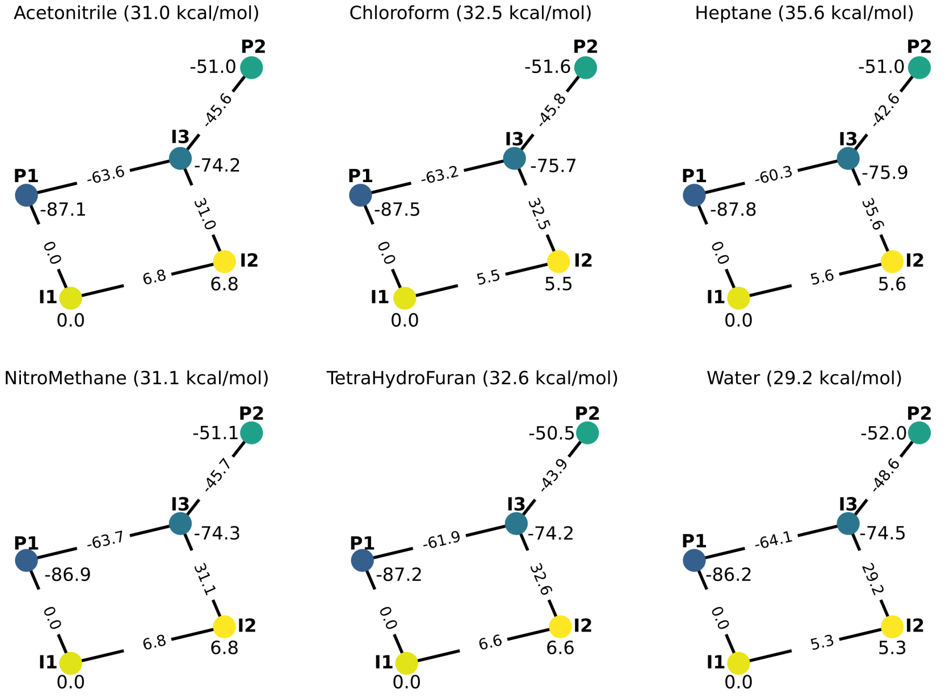


Figure 5.14: Reaction network graphs for peroxyformate decomposition in a selection of six solvents (acetonitrile, chloroform, heptane, nitromethane, tetrahydrofuran and water). Energies are Gibbs free energies, in $\mathrm{kcal \cdot mol^{-1}}$. The overall free energy reaction barrier for each solvent is given between parenthesis after each solvent name.

query are collected in Table 5.1.

5.4.2 Complex reaction networks: the decomposition of indole

The growing relevance of automated approaches to mechanistic searches and the subsequent complications of the management and interpretation of the produced reaction networks was thoroughly discussed along Chapter III, presenting the amk-tools library to process and visualize the networks generated by AutoMeKin. Moreover, apart from the main interactive visualization module, the code also provided a direct interface between

Table 5.1: Query results for Code example 5.2 in the peroxyformate decomposition KG. Contains stage identifiers of pairwise-connected nodes and the corresponding TS when applicable, and None elsewhere.

Node	Node	TS
stg-4	stg-0	stg-5
stg-2	stg-0	stg-6
stg-0	stg-1	stg-7
stg-2	stg-3	None
stg-3	stg-4	None

AutoMeKin results and ioChem-BD, uploading the raw computational results and the network structure altogether.

This feature matches perfectly our current workflow (Figure 5.4), defining the collection of data and its accompanying network structure in a single step. Therefore, the knowledge graph generation protocol can be directly used for automatically discovered networks (Figure 5.15), opening the door to the creation of complex knowledge graphs with minimal user intervention both for the creation and the manipulation of the target data.

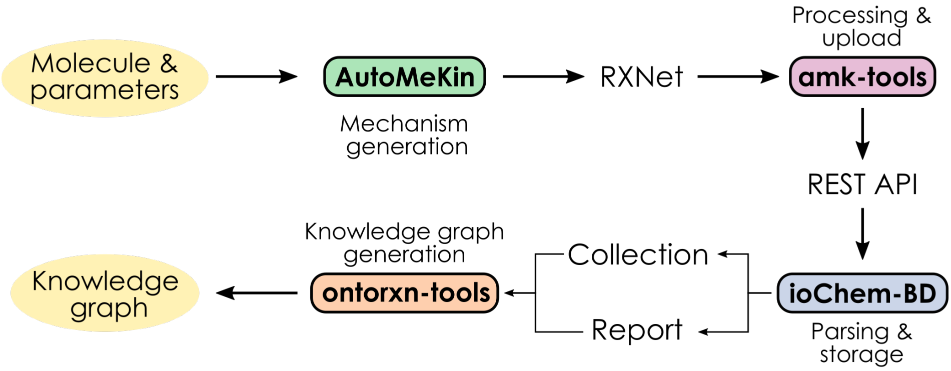


Figure 5.15: Scheme of the program pipeline employed to build knowledge graphs from the RXNets generated by AutoMeKin

No further operational considerations need to be made, with the upper part of the workflow in Fig. 5.15 being just as explained in Chapter III and the lower part being already discussed in Section 5.3. Thus, we may directly proceed to a specific application example, considering the same indole decomposition network that we presented on Chapter III, with 72 different ChemSpecies, 67 NetworkStages and 40 ReactionSteps. From this point of view, the underlying network is remarkably larger and more interconnected than in the peroxyformate example, while also presenting a much simpler 1:1 mapping between species and calculations. As computational studies on reaction networks can showcase widely different kinds of complexity, the organization approach that we propose with OntoRXN should be able to accommodate all these situations, as demonstrated here.

As both the chemistry of indole decomposition and the details of KG generation have been already introduced, it just remains to provide a couple of examples of how SPARQL querying can be applied to this specific system. One question of interest that can be very easily answered through this approach is to determine how many times are molecular fragments repeated across the network, giving an idea of how many different fragmentation schemes lead to the same species (Code example 5.3).

```
PREFIX rxn: <http://www.semanticweb.com/OntoRxn#>
SELECT DISTINCT ?spcX (COUNT(?stgX) AS ?Ncount)
  (SAMPLE(?labX) AS ?lab)
  (GROUP_CONCAT(?nameX) AS ?stages)
WHERE {
  ?stgX rxn:hasSpecies ?spcX .
  ?spcX rxn:hasCalculation ?calcX .
  ?calcX rxn:hasAnnotation ?labelX .
  BIND(STRBEFORE(?labelX,’;’) AS ?labX)
  OPTIONAL{?stgX rxn:hasAnnotation ?nameX}
}
GROUP BY ?spcX
ORDER BY DESC(?Ncount)
LIMIT 5
```

Code example 5.3: SPARQL query to determine the number of stages in which each species in the system appears and their corresponding names, returning only the five top results ordered by occurrence number.

Keeping on with the idea of clarifying SPARQL keywords and functions, Code example 5.3 introduces a new agreggating function, **COUNT**, which just returns the number of occurrences of the target variable in the group, and the **BIND** keyword employed to create named variables during the query evaluation. In this case, this goes together with the string manipulation function **STRBEFORE** which grabs the part of a string variable going before a specified separator. Finally, **ORDER BY** simply reorders the result table according to a given variable, which in this case is the number of stages in which a specific species participates, and **LIMIT** selects only the top five results from the query, shown in Table 5.2.

Table 5.2: Number of occurrences per network stage of the most common fragments in the indole network and corresponding stage names selecting the top five results only.

Frag.	No. stages	Sel. stages
CN	6	PR342 PR313 PR315 PR136 PR320 PR409
HCN	2	PR3 PR101
HNC	2	PR120 PR278
PR155	1	PR155
CH2	1	PR155

From Table 5.2, we have that only three of the fragments in the network are shared by different stages, with cyanide radical being the most common participating in six stages and the isomeric HCN and HNC structures appearing in two. While for this mid-sized reaction network the insights provided by the query in Code example 5.3 are not particularly relevant, the advantage of this approach is that the very same query may be directly applied to any other knowledge graph, streamlining the exploration of complex systems and eventually providing an additional functionality to the amk-tools workflow.

Finally, following the graph regeneration stategy from Code example 5.2, we can easily recover the reaction network and map any property from the KG. Among the available properties, we can take the InChI associated to the ChemSpecies contained in every stage, which are parsed in ioChem-BD and fed to the knowledge graph. InChIs provide an unique molecular identifier for every molecular entity in the network, which can be used to, for example, generate their corresponding 2D molecular representations through the

RDKit library [54] and embed them in the network depiction, allowing for a simple generation of traditional-mechanism-like schemes for automatically discovered reaction networks (Figure 5.16).

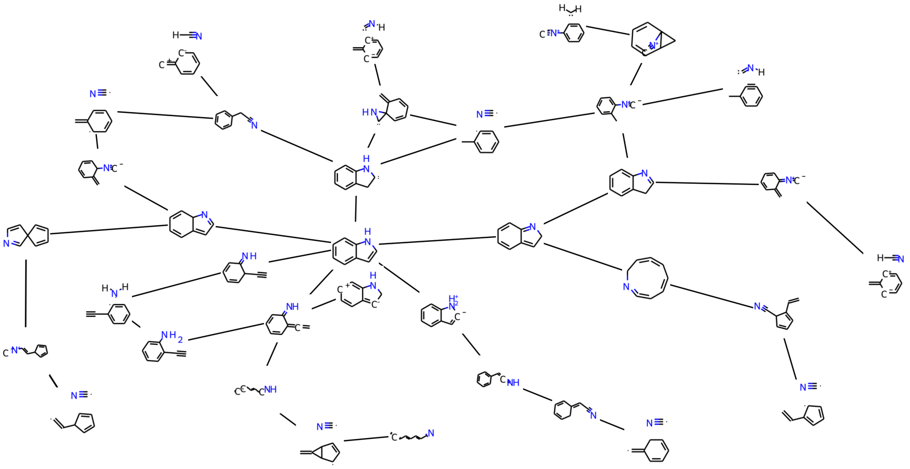


Figure 5.16: Reaction network graphs for indole decomposition, with molecular depictions for every node taken from InChI strings.

5.4.3 Utilization of KGs in complex simulation workflows: stereoselective carbon dioxide fixation

Through the two previous examples (peroxyformate decomposition and indole decomposition), we employed the corresponding KGs as a tool to simplify the analysis of the data that was already encoded in the network, extracting energy profiles, activation energies, connectivity information, molecular string identifiers, etc. However, another aim of the semantic approach is to present knowledge graphs as a standard format to pass reaction networks to other calculation tools. Thus, KGs would effectively become a piece on the development of automated workflows, retrieving only the properties of interest through standardized SPARQL queries and minimizing the need for manual interaction along the processing pipeline.

To demonstrate this feature, we built a workflow to generate and

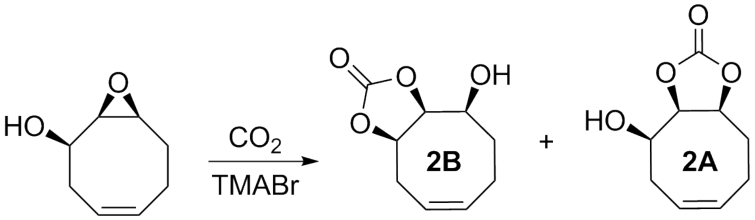


Figure 5.17: CO_2 fixation reaction over cyclooctene epoxy alcohol derivative, including main diastereoisomeric cyclic carbonate product 2B and the minor product 2A.

run a microkinetic model directly from a knowledge graph, automating the detection of the chemical equations giving rise to the model and the computation of the corresponding rate constants from the activation free energies through the Eyring equation. As a target system, we considered the stereospecific fixation of carbon dioxide on a cyclic epoxyalcohol derivative, catalyzed by a bromide salt (Figure 5.17), which was observed to produce a single major diastereoisomeric cyclic carbonate [243, 244]. The complete reaction scheme including computed Gibbs free energies from DFT computations[3] is depicted in Figure 5.18.

Here, the flexibility of the cyclooctene ring generates a non-trivial landscape, as not only both the α and β positions of the epoxide substrate EpOr can be attacked by the nucleophile, but also there can be a proton transfer between the alkoxide group on position C_6 and the hydroxo group at C_4 when the attack occurs in α. This process is deeply facilitated by the spatial disposition of the cyclooctene core, which allows the two oxygen atoms and the hydrogen to form a strong hydrogen bond leading to a pseudo-6-cyclic structure. Through this manifold, the isomerized α-alkoxide (AMAK) can produce an isomeric epoxide (EpIsom) which is only 1.5 $\text{kcal} \cdot \text{mol}^{-1}$

[3] Computational details from Ref. [243]. Unrestrained geometry optimizations were carried out using Gaussian09 [238] at the B97D3 [245, 246] level of theory, with a 6-311G(d,p) [240, 241] basis set and employing the SMD [242] model for implicit solvation with the default parameters for butanone. Minima and saddle points were identified by harmonic vibrational frequency analysis, ensuring the presence of 0 or 1 imaginary frequencies, respectively. Dataset collection available in ioChem-BD [244].

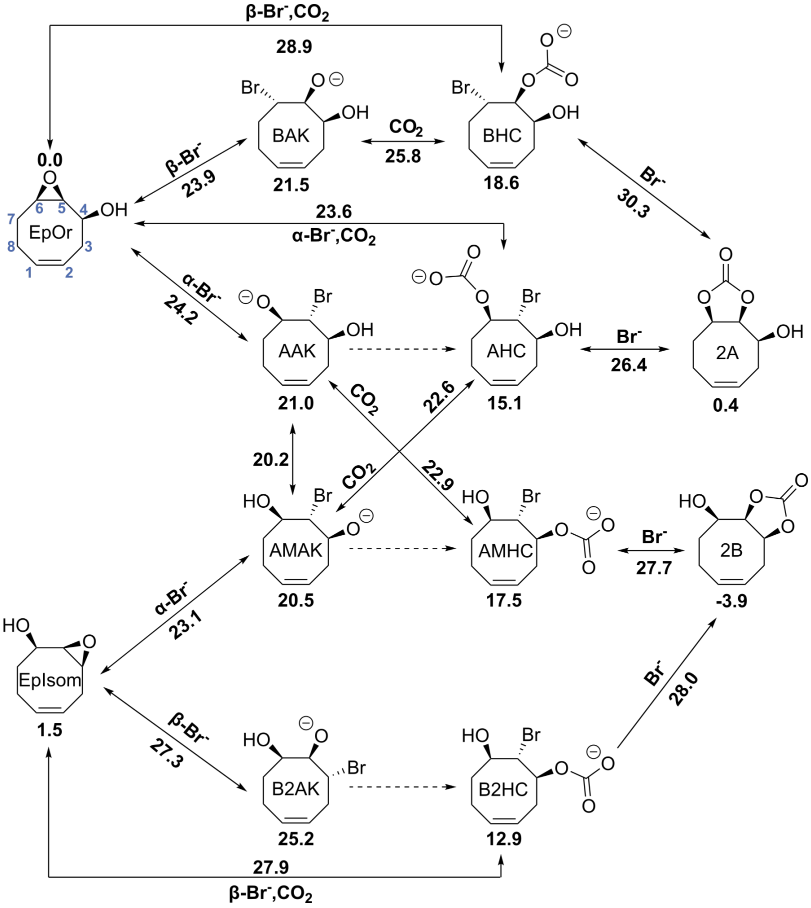


Figure 5.18: DFT-characterized mechanism for cyclic carbonate formation. Gibbs free energies, in kcal · mol^{-1}, at a reference state of 1.0 M and 353.15 K, at the B97D3/6-311G(d,p) level with SMD parameters for butanone as a solvent. The counterion (tetramethylammonium or TMA) is not depicted in the scheme and is not included in the intermediate characterization, but is considered in the energy reference (TMABr + CO_2 + EpOr). Dashed arrows show expected transformations for which a TS could not be characterized.

above the original one. Moreover, we were also able to characterize several processes in which the nucleophilic attack and the fixation of CO_2 are concerted, going directly from the epoxide to the hemicarbonate. All of these routes are strongly intertwined and quite close in energy, with barriers ranging from 24 to 30 $\mathrm{kcal \cdot mol^{-1}}$, with 2B being the thermodynamic product and 2A the kinetic one. Due to all these complications, the direct analysis of the mechanism in terms of its energies is problematic, prompting us to plan a microkinetic simulation to try to reproduce the conversion and selectivity values registered in the experiments.

From this mechanism, we generated the knowledge graph following the same protocol as in the previous situations, obtaining a graph with 12 steps, 28 stages and 32 species and calculations. To build the microkinetic model, we need to pay attention to two main aspects: the generation of the stoichiometry of the corresponding reactions and the recalculation of the Gibbs free energies to change the reference state from the standard 1 atm, 298.15 K state obtained from the calculations (gas phase, standard temperature) to the 1.0 M state for processes in solution and the working temperature of 353.15 K. This reference state switch from the gas phase to the solution is essential for a proper application of microkinetic modeling to homogeneous systems [151], while the modification of the reference temperature also allows to improve the concordance with experimental conditions. The recalculation of Gibbs free energies just implies the recalculation of the partition functions for all the calculations in the system, employing standard formulas from statistical thermodynamics. Here, the structure of the KG greatly simplifies the task of extracting the necessary parameters from the network.

On the other hand, the determination of the individual reactions encoded in the KG (and required to setup the microkinetic model) follows directly from the ontology structure, as every elementary reaction is already defined as a ReactionStep entity. Thus, it is only necessary to determine the ChemSpecies that are being transformed at every ReactionStep, retrieving them from the two nodes that are associated to the step. Due to the atom

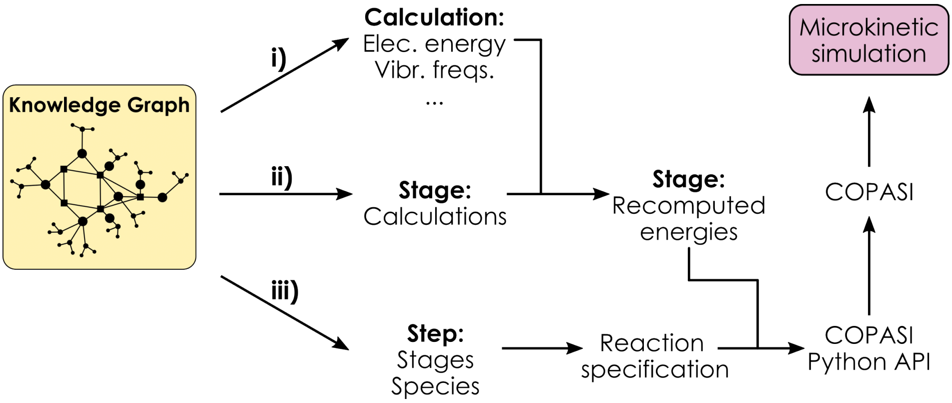


Figure 5.19: Schematic depiction of the querying workflow to set up a microkinetic model in COPASI from the knowledge graph.

consistency that is enforced at the NetworkStage level to properly compute relative energies, the chemical equations determined from ReactionSteps will be immediately balanced. From all this information, the actual model will be generated and run through the COPASI [140] program, with all reactions being assumed as reversible, mass-action governed processes, calculating rate constants from relative barriers through the Eyring equation. In this way, no direction of network traversal needs to be assigned, as the own simulation marks the chemical flow of the system.

Thus, the overall model setup involves a set of three SPARQL queries, as highlighted in the workflow scheme in Figure 5.19. The protocol starts, as aforementioned, by fetching all required properties for partition function recalculation through the query **i**, which is presented in Code example 5.4. Through these results, Gibbs free energies for all CompCalculation entities can be recomputed at the requested values of pressure and temperature, effectively modifying the reference state of the calculations ot our desired conditions, From there, query **ii** (Code example 5.5) allows to match stages with the calculations that they comprise, as done in the previous examples.

The combination of **i)** and **ii)** allows to effectively obtain recomputed Gibbs free energies for all stages (intermediates and transition states) across the reaction network, which will be used to compute the relative barriers for

```
PREFIX rxn: <http://www.semanticweb.com/OntoRxn#>
PREFIX gc: <http://purl.org/gc/>
SELECT DISTINCT ?calcX ?Eel ?molmass ?moi ?symmnumb ?freqlist
WHERE {
  ?calcX gc:hasResult / rxn:hasElecEnergy / gc:hasValue ?Eel .
  OPTIONAL{?calcX gc:hasResult / rxn:hasMolMass / gc:hasValue
      ?molmass} .
  OPTIONAL{?calcX gc:hasResult / rxn:hasMomentInertia / gc:
      hasValue ?moi} .
  OPTIONAL{?calcX gc:hasResult / rxn:hasFreqList / gc:hasValue
       ?freqlist} .
  OPTIONAL{?calcX rxn:hasSymmetryNumber ?symmnumb}
  }
ORDER BY ?calcX
```

Code example 5.4: SPARQL query to fetch descriptors for partition function and thermodynamic magnitude recalculation, including electronic energy, molecular mass, moment of inertia, symmetry number and vibrational frequencies, for every *CompCalculation* object in the KG.

```
  PREFIX rxn: <http://www.semanticweb.com/OntoRxn#>
  PREFIX gc: <http://purl.org/gc/>
  SELECT DISTINCT ?stgX (SAMPLE(?nameX) AS ?name)
    (GROUP_CONCAT(?calcX ; separator=';') AS ?calcList)
  WHERE {
    ?stgX rxn:hasSpecies ?spcX .
    ?spcX rxn:hasCalculation ?calcX .
    OPTIONAL {?stgX rxn:hasAnnotation ?nameX}
  }
  GROUP BY ?stgX
  ORDER BY ?stgX
```

Code example 5.5: SPARQL query to map *NetworkStage* entities to their corresponding calculations, assuming 1:1 mapping between species and calculations.

every step in the network. Then, the query **iii** (Code example 5.6) defines all unique reactions in the network from the corresponding ReactionStep entities.

Query **iii** fetches the two intermediates (as NetworkStage entities) connected by a given ReactionStep, and then determines the set of species that belong to that stage. Then, the names of these species, defined when generating the reaction network in ioChem-BD, are combined to define the two sides of the corresponding elementary reaction. Additionally, the

```
PREFIX rxn: <http://www.semanticweb.com/OntoRxn#>
PREFIX gc: <http://purl.org/gc/>
SELECT DISTINCT ?stepX stgX ?stgTS
        (GROUP_CONCAT(?spcName ; separator=’+’) as ?spcNode)
WHERE {
  ?stepX rxn:hasNode ?stgX .
  ?stgX rxn:hasSpecies ?spcX .
  ?spcX rxn:hasCalculation ?calcX .
  ?calcX rxn:hasAnnotation ?noteX .
  BIND(STRBEFORE(?noteX,’;’) AS ?spcName) .
  ?stepX rxn:hasTS ?stgTS
}
GROUP BY ?stepX ?stgX ?stgTS
ORDER BY ?stepX
```

Code example 5.6: SPARQL query to fetch reaction specifications from the KG, mapping every *ReactionStep* to the corresponding stages and the *ChemSpecies* belonging to them.

identifiers of the stages for the intermediates and for the transition state are also retrieved, so the results of this query can be used to directly define the reactions, mapping the identifiers with the previously computed energies.

At this point, all the information required for the model has been already extracted from the knowledge graph. It just remains to clean up the reaction stoichiometries, removing species appearing at both sides of the reaction (which are unaffected by that particular process), and to compute the relative barriers in the two possible traversal directions of the step. Then, the complete set of reactions encoded in the KG is finally defined and the relative barriers for the individual reactions can be transformed to rate constants, feeding all this data to COPASI to finally run the microkinetic model. At this point, only conditions such as initial reactant concentrations, simulation time and temperature remain to be specified to run the simulation. Thus, to be as close as possible to the experimental conditions, we set the following parameters:

T	[EpOr]	P(CO_2)	[TMABr]
353.15 K	2.5 M	40 bar	10 - 25% mol

To determine the concentration of CO_2 in the liquid phase, we considered the results reported by Sato et al. [247] for the solubility of carbon dioxide in 2-butanone at 353 K, our working temperature, and extrapolated the reported pressure/molar fraction diagrams to the 40 bar employed in the experimental setup, obtaining $x(\mathrm{CO_2}, 40\,\mathrm{bar}) = 0.3258$. From there, we can obtain CO_2 concentration from the expression for the molar fraction, taking solvent concentration as its molar volume. At 353 K, the reported density of butanone is 747 g cm^{-3} [248], giving $[MEK] = n/V = \rho/M_w = 10.4$ M. From there, CO_2 concentration is estimated as 4.98 M, thus being in clear excess compared to the epoxide which is the limiting reagent. In general, this concentration estimation is not trivial: we may introduce the concentrations of epoxide and catalyst also in the molar fraction, as they are not negligible, but this will prompt us to decide whether we assume the molar fraction from [247] is changed in the more complex solution or not. Thus, for simplicity, we carried out the simulations with the $[CO_2] = 4.98$ M estimation neglecting the co-solutes.

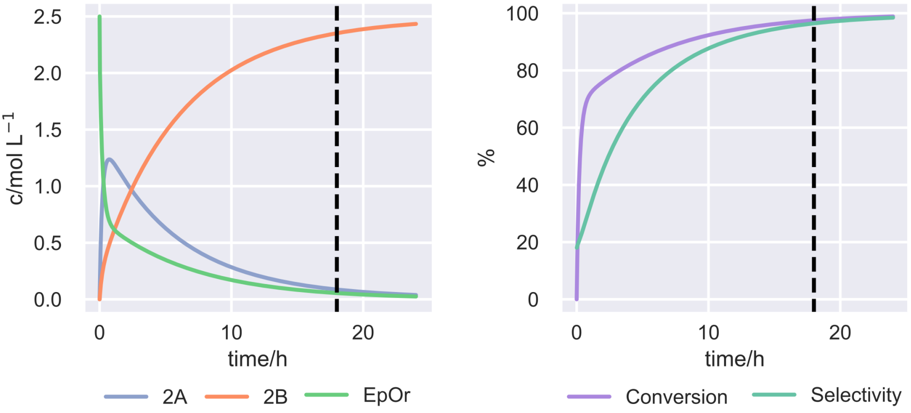


Figure 5.20: Results of the microkinetic model for cyclic carbonate production. Left, concentration vs. time plot for the reactant (EpOr) and the two possible products 2A and 2B. Right, time evolution of reactant conversion and product selectivity. Dashed lines at t = 18 h are shown to mark the time of finalization of the experiments, for comparison.

The microkinetic simulation results for [TMABr] = 25% mol = 0.625 M case are collected in Figure 5.20, which shows how the process is under thermodynamic control: although initially the kinetic product 2A is formed faster, the equilibria are then reversed and almost most of the pre-formed 2A is transformed to 2B at t = 18 h, which was the experimental reaction time. In the plot on the right, we see a very quick conversion, reaching 80% during the first hour of reaction, while the selectivity for 2B formation increases more slowly as the $2A \to 2B$ transformation takes place. All in all, computed results are in very good agreement with the experiments, allowing us to compare the registered conversions and selectivities for different initial loadings of the bromide salt catalyst (Table 5.3), showing a good agreement between the computed results and the experiments.

	Model		Experiment	
$[Br^-]$/M	Conv.	Sel.	Conv.	Sel.
0.625	98	96	99	92
0.5	96	94	90	87
0.375	93	89	85	89
0.25	89	80	85	88

Table 5.3: Comparison of predicted and experimental conversions and selectivities for 2B formation with catalyst loadings ranging from 10% to 25 % mol.

Developing workflows based on knowledge graphs has the major advantage of their *transferrability*: in principle, the complete simulation protocol that we have outlined for this cyclic carbonate formation could be directly applied to any other reaction network expressed as a KG. Having this kind of standardized format bringing together all the information on a given reactive system simplifies task automation by allowing to skip steps which are often quite time-consuming such as parsing or organizing the information from the raw computational outputs.

Overall, we have presented several relevant SPARQL queries along the section, which might serve as a template to build further queries on either the example datasets we have utilized or any other CRN uploaded to ioChem-BD. To give a better idea of the functionality of these queries and also make our knowledge graphs available, we set up a web service [249] providing a

RDF database and SPARQL endpoint based on the Blazegraph [250] engine. Through this service, all the specific SPARQL queries discussed along the chapter (Code examples 5.1 to 5.6) can be read, modified and run against the peroxyformate, indole and cyclic carbonate knowledge graphs. The idea behind this demonstration goes along with one of the main principles followed along this Thesis, which is not only to create new tools and services to manage reaction networks, but also to make them as accessible and user-friendly as possible. Thus, we believe that providing this interactive frontend to our ontology can help with the dissemination of our semantic approach to reaction networks along the community, facilitating a wider adoption of the proposed methodology so it can eventually become a true data management standard.

5.5 Conclusions and future work

We have proposed a semantic organization scheme for chemical reaction networks characterized from computational studies, developing the OntoRXN ontology to provide the terms and structure required to apply this paradigm. Along this first implementation, we have aimed for a proof-of-concept approach, focusing on the core ontology structure and its connection with the ioChem-BD database through the knowledge graph instantiation agents available on the ontorxn-tools library. Moreover, we have also proposed several post-processing agents to demonstrate the interest of knowledge graphs as a standard reaction network format that can be easily integrated into data analysis or simulation pipelines. These agents have been employed to process the knowledge graphs corresponding to three distinct reaction mechanisms in different manners, showcasing the versatility of the semantically-organized reaction networks. The first application example involved the simple generation of reaction energy profiles and the characterization of reaction barriers for the decomposition of tert-butyl peroxyformate, characterized in a wide selection of implicit solvents. Then, an automatically discovered network for indole decomposition was used to depict a graph including molecular 2D representations at every node, which are built from the InChI descriptors included in the knowledge graph. Finally, a protocol for an automated setup of microkinetic simulations was illustrated from a mechanism involving the fixation of CO_2 over a cyclooctene oxide derivative, utilizing the KG to compute Gibbs free energies at requested values of pressure and temperature and to determine the individual reactions encoded in the networks with their direct and reverse rate constants, as required for this kind of simulations. Through these examples, we were able to demonstrate the interest of the ontology-based approach to handle chemical data with ease for a range of non-trivial applications.

Building from this initial proposal, several avenues for developing and utilizing OntoRXN and its knowledge graphs can be thought of. An obvious first point to tackle is the extension of the ontology, adding properties and classes to manage the plethora of fields that ioChem-BD captures but are not yet defined at the ontology end. Of course, apart from OntoRXN itself, the interface with the CML files in the database via XSL stylesheets also needs to be grown accordingly. In the long term, it shall be possible to have a 1:1 mapping between all the properties captured by ioChem-BD and the OntoRXN ontology. Moreover, the interface itself, currently involving a external Python library, shall eventually be fully integrated into ioChem-BD, so KGs could be directly generated from the database itself to simplify the overall pipeline.

Another important aspect regarding the evolution of the ontology is the identification and addition of novel connections with other chemical ontologies, as hinted along Sections 5.1 and 5.2, following the integrative and collaborative spirit of the Semantic Web. For example, we have already proposed the interest of linking the reaction steps defined in our framework with the named elementary processes appearing on the MOP ontology. A way to actually drive this aspect forward would be to create agents capable of automatically identifying the reaction types encoded in a given step to automatically assign them to the classes defined in MOP.

Focusing more on the applicability of the current toolkit, it will also be interesting to generate a larger set of knowledge graphs for reaction mechanisms, building a database for KG-organized reaction networks. This likely large and diverse, but strongly structured, dataset could be an ideal target to automated analysis and Machine Learning techniques.

Chapter VI

Conclusions

Throughout this Thesis we have explored different approaches to improve the understanding and management of complex chemical reaction networks (CRNs) constructed from the information gathered through computational mechanistic discovery. In this way, we put the spotlight on networks as an ideal representation for intricate mechanisms, according to their versatility on modeling and the immediacy with which they allow the application of Graph Theory techniques and concepts to chemical problems. Nevertheless, we cannot disregard alternate representations such as individual reactions or energy profiles (as described in Chapter II), which also have clear advantages in terms of chemical interpretation. Thus, another important part in our framework came to be the interchangeability of all these ways to describe chemical reactivity, with the main aim of choosing the representation that best fits a given job in every case. These principles were then applied to the development of the different software pieces that are introduced along the Thesis, tackling multiple facets and stages of CRN characterization and analysis. Moreso, the whole toolkit is connected with the ioChem-BD repository aiming for a synergystic relationship: the database provides a central point for data storage and retrieval, with an unified data format, and the newly developed tools prompt and provide novel functionalities to the ioChem-BD platform.

To sum up, we will summarize the main conclusions extracted from the three main results' chapters of the Thesis (Chapters III, IV and V): while we will collect the most important aspects here, we remind the reader that more detailed conclusions and some future perspectives on each of the described projects are provided at the end of each of these individual chapters.

Chapter III shows the interest of providing filtering and visualization frameworks to treat the large and intertwined CRNs generated by automated mechanism discovery tools, presenting the interactive dashboards that **amk-tools** produces directly from the results of AutoMeKin. The application of the toolkit to indole decomposition allows a simple and user-friendly reproduction and refinement of previous DFT-computed profiles with minimal effort, together with the discovery of alternative channels that were overlooked in those previous studies. The reaction pathways discovered through the combination of AutoMeKin and amk-tools provided additional knowledge on the possible roles of indole in two widely different areas. Regarding coal tar pyrolysis, several new routes leading to hydrogen cyanide and hydrogen isocyanide could be reported, as well as a refinement of undercharacterized channels forming amino radical. In the context of astrochemical reaction networks, we proposed a feasible pathway for the barrierless production of indole starting from methylene radical and phenyl isocyanide. Additionally, by directly including AutoMeKin results in the ioChem-BD database, bridging together the raw computational data with its chemical contextualization (the reaction network) it becomes possible to have a more complete representation of these complex chemical system inside the database, following the principles of Findable, Accessible, Interoperable and Recyclable (FAIR) data.

Chapter IV shows the strengths of the graph-based energy span model (ESM) implemented in **gTOFfee** to treat complex catalytic cycles including off-cycle species and intertwined mechanisms, effectively extending the applicability of the model as an out-of-the-box tool to extract chemically relevant information from DFT or *ab initio* results. We also introduce the *effective energy span*, computed from the TOF, as a powerful descriptor for the apparent activation energy of such systems. The first application example, tackling the well-known hydroformylation of ethylene over cobalt catalysts, shows the promising capabilities of gTOFfee for homogeneous systems. In this way, we were able to produce selectivity maps exploring different initial concentrations of reactants and products that were found to be in good agreement with experimental results and microkinetic simulations. The final part of the chapter delves on the applicability of the tool to heterogeneous catalysis, demonstrating how the graph-based approach is able to overcome several of the key limitations that the linear-based ESM presents for this kind of systems, although still leaving quite some room for improvement.

Chapter V illustrates the versatility of semantically organizing chemical reaction networks (producing *knowledge graphs*) by means of the **OntoRXN** ontology, presenting multiple real-world examples where the strong data structure enforced by the ontology enables a simple and robust analysis and reorganization of the abundant data generated by the underlying calculations. The resulting chemical reaction network knowledge graphs (CRN-KGs) permit an easy extraction of the information of interest through simple and readable SPARQL queries, owing to the enforced structure of the semantic-based data model. From the chemical viewpoint and along our set of test reaction mechanisms, we were able to extract the reaction energy profiles and free energy barriers for a complex dataset involving the characterization of the mechanism of peroxyformate decomposition in a wide variety of implicit solvents. Moreover, we could also generate enhanced graph depictions for the indole decomposition network including the corresponding molecular 2D representations for every node in the network. Finally, we built a pipeline for automatically setting up and performing microkinetic simulations over any CRN-KG, applying it to a CO_2 fixation mechanism on a cyclooctene oxide derivative to extract the corresponding chemical reactions, activation free energies and rate constants. As a final note, the direct connection of the generation of CRN-KGs with the ioChem-BD database does not only greatly simplify the production stage, as all required connectivity information and results are already present on the database, but does also provide a powerful new layer of functionality to the ioChem-BD platform itself.

Bibliography

[1] P. Muller, *Pure and Applied Chemistry* **1994**, *66*, 1077–1184, DOI 10.1351/pac199466051077 (p. 6, 8).

[2] C. A. Busacca, D. R. Fandrick, J. J. Song, C. H. Senanayake, *Advanced Synthesis & Catalysis* **2011**, *353*, 1825–1864, DOI 10.1002/adsc.201100488 (p. 8).

[3] F. Poovan, V. G. Chandrashekhar, K. Natte, R. V. Jagadeesh, *Catalysis Science & Technology* **2022**, -, DOI 10.1039/D2CY00232A (p. 8).

[4] Intergovernmental Panel on Climate Change, *Climate Change 2021: The Physical Science Basis. Contribution of Working Group I to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change*, (Eds.: V. Masson-Delmotte et al.), Cambridge University Press, Cambridge, United Kingdom and New York, NY, USA, **2021** (p. 10).

[5] Intergovernmental Panel on Climate Change, *Climate Change 2022: Impacts, Adaptation, and Vulnerability. Contribution of Working Group II to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change*, (Eds.: H.-O. Pörtner, D. Roberts, M. Tignor, E. Poloczanska, K. Mintenbeck, A. Alegría, M. Craig, S. Langsdorf, S. Löschke, V. Möller, A. Okem, B. Rama), Cambridge University Press, **2022** (p. 10).

[6] Intergovernmental Panel on Climate Change, *Climate Change 2022: Mitigation of Climate Change. Contribution of Working Group III*

to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change, (Eds.: P. Shukla, J. Skea, R. Slade, A. A. Khourdajie, R. van Diemen, D. McCollum, M. Pathak, S. Some, P. Vyas, R. Fradera, M. Belkacemi, A. Hasija, G. Lisboa, S. Luz, J. Malley), Cambridge University Press, Cambridge, UK and New York, NY, USA, **2022**, DOI 10.1017/9781009157926 (p. 10).

[7] J. W. Erisman, M. A. Sutton, J. Galloway, Z. Klimont, W. Winiwarter, *Nature Geoscience* **2008**, *1*, 636–639, DOI 10.1038/ngeo325 (p. 10).

[8] C. Smith, A. K. Hill, L. Torrente-Murciano, *Energy & Environmental Science* **2020**, *13*, 331–344, DOI 10.1039/C9EE02873K (p. 10).

[9] J. Humphreys, R. Lan, S. Tao, *Advanced Energy and Sustainability Research* **2021**, *2*, 2000043, DOI 10.1002/aesr.202000043 (p. 10).

[10] J. K. Nørskov, T. Bligaard, A. Logadottir, J. R. Kitchin, J. G. Chen, S. Pandelov, U. Stimming, *Journal of The Electrochemical Society* **2005**, *152*, J23, DOI 10.1149/1.1856988 (p. 10).

[11] I. Roger, M. A. Shipman, M. D. Symes, *Nature Reviews Chemistry* **2017**, *1*, 0003, DOI 10.1038/s41570-016-0003 (p. 10).

[12] Nobel Prize Outreach AB, The Nobel Prize in Chemistry 2021, **2021**, https://www.nobelprize.org/prizes/chemistry/2021/summary/ (visited on Sept. 11, 2022) (p. 10).

[13] M. Aresta, A. Dibenedetto, A. Angelini, *Chemical Reviews* **2014**, *114*, 1709–1742, DOI 10.1021/cr4002758 (p. 11).

[14] Q. Liu, L. Wu, R. Jackstell, M. Beller, *Nature Communications* **2015**, *6*, 5933, DOI 10.1038/ncomms6933 (p. 11).

[15] G. Centi, S. Perathoner, A. Salladini, G. Iaquaniello, *Frontiers in Energy Research* **2020**, *8*, 567986, DOI 10.3389/fenrg.2020.567986 (p. 11).

[16] T. Engel, *Journal of Chemical Information and Modeling* **2006**, *46*, 2267–2277, DOI 10.1021/ci600234z (p. 11).

[17] S. Chmiela, H. E. Sauceda, K.-R. Müller, A. Tkatchenko, *Nature Communications* **2018**, *9*, 3887, DOI 10.1038/s41467-018-06169-2 (p. 12).

[18] H. E. Sauceda, S. Chmiela, I. Poltavsky, K.-R. Müller, A. Tkatchenko, *The Journal of Chemical Physics* **2019**, *150*, 114102, DOI 10.1063/1.5078687 (p. 12).

[19] O. T. Unke, S. Chmiela, H. E. Sauceda, M. Gastegger, I. Poltavsky, K. T. Schütt, A. Tkatchenko, K.-R. Müller, *Chemical Reviews* **2021**, *121*, 10142–10186, DOI 10.1021/acs.chemrev.0c01111 (p. 12).

[20] H. Eyring, *The Journal of Chemical Physics* **1935**, *3*, 107–115, DOI 10.1063/1.1749604 (p. 14).

[21] A. Szabo, N. S. Ostlund, *Modern Quantum Chemistry: Introduction to Advanced Electronic Structure Theory*, Dover Publications, Mineola, N.Y, **1996** (p. 15).

[22] F. Jensen, *Introduction to Computational Chemistry*, 2nd ed, John Wiley & Sons, Chichester, England ; Hoboken, NJ, **2007** (p. 15).

[23] P. Hohenberg, W. Kohn, *Physical Review* **1964**, *136*, B864–B871, DOI 10.1103/PhysRev.136.B864 (p. 16).

[24] W. Kohn, L. J. Sham, *Physical Review* **1965**, *140*, A1133–A1138, DOI 10.1103/PhysRev.140.A1133 (p. 16).

[25] J. P. Perdew, K. Schmidt in AIP Conference Proceedings, *Vol. 577*, AIP, Antwerp (Belgium), **2001**, pp. 1–20, DOI 10.1063/1.1390175 (p. 17).

[26] G. E. Moore, *IEEE Solid-State Circuits Society Newsletter* **2006**, *11*, 33–35, DOI 10.1109/n-ssc.2006.4785860 (p. 17).

[27] G. E. Moore, *IEEE Solid-State Circuits Society Newsletter* **2006**, *11*, 36–37, DOI 10.1109/N-SSC.2006.4804410 (p. 17).

[28] D. Rotman, *MIT Technology Reviews* **2020** (p. 18).

[29] A. Jain, S. P. Ong, W. Chen, B. Medasani, X. Qu, M. Kocher, M. Brafman, G. Petretto, G.-M. Rignanese, G. Hautier, D. Gunter, K. A. Persson, *Concurrency and Computation: Practice and Experience* **2015**, *27*, 5037–5059, DOI 10.1002/cpe.3505 (p. 19, 50).

[30] E. E. Schadt, M. D. Linderman, J. Sorenson, L. Lee, G. P. Nolan, *Nature Reviews Genetics* **2010**, *11*, 647–657, DOI 10.1038/nrg2857 (p. 20).

[31] S. Sagiroglu, D. Sinanc in 2013 International Conference on Collaboration Technologies and Systems (CTS), IEEE, San Diego, CA, USA, **2013**, pp. 42–47, DOI 10.1109/CTS.2013.6567202 (p. 20).

[32] M. Hutson, *Science* **2019**, *365*, 416–417, DOI 10.1126/science.365.6452.416 (p. 20).

[33] Y.-C. Lo, S. E. Rensi, W. Torng, R. B. Altman, *Drug Discovery Today* **2018**, *23*, 1538–1546, DOI 10.1016/j.drudis.2018.05.010 (p. 20).

[34] P. Schlexer Lamoureux, K. T. Winther, J. A. Garrido Torres, V. Streibel, M. Zhao, M. Bajdich, F. Abild-Pedersen, T. Bligaard, *ChemCatChem* **2019**, *11*, 3581–3601, DOI 10.1002/cctc.201900595 (p. 20).

[35] G. dos Passos Gomes, R. Pollice, A. Aspuru-Guzik, *Trends in Chemistry* **2021**, *3*, 96–110, DOI 10.1016/j.trechm.2020.12.006 (p. 20).

[36] K. M. Jablonka, D. Ongari, S. M. Moosavi, B. Smit, *Chemical Reviews* **2020**, *120*, 8066–8129, DOI 10.1021/acs.chemrev.0c00004 (p. 20).

[37] E. Martínez-Núñez, G. L. Barnes, D. R. Glowacki, S. Kopec, D. Peláez, A. Rodríguez, R. Rodríguez-Fernández, R. J. Shannon, J. J. Stewart, P. G. Tahoces, S. A. Vazquez, *Journal of Computational Chemistry* **2021**, *42*, 2036–2048, DOI 10.1002/jcc.26734 (p. 22).

[38] S. Kozuch, S. Shaik, *Accounts of Chemical Research* **2011**, *44*, 101–110, DOI 10.1021/ar1000956 (p. 22, 87, 92).

[39] D. B. West, *Introduction to Graph Theory, 2nd Edition*, Second, Prentice Hall, **2001** (p. 29).

[40] R. J. Wilson, *Introduction to Graph Theory, 4th Edition*, Fourth, Addison Westley Longman Ltd., **1996** (p. 29).

[41] J. L. Gross, J. Yellen, M. S. Anderson, *Graph Theory and Its Applications, 3rd Edition*, 3rd, CRC Press, **2019** (p. 29, 35).

[42] E. W. Dijkstra, *Numerische Mathematik* **1959**, *1*, 269–271, DOI 10.1007/BF01386390 (p. 33).

[43] A. T. Balaban, *Journal of Chemical Information and Computer Sciences* **1985**, *25*, 334–343, DOI 10.1021/ci00047a033 (p. 36, 37).

[44] A. T. Balaban, *Journal of Chemical Information and Computer Sciences* **1995**, *35*, 339–350, DOI 10.1021/ci00025a001 (p. 36).

[45] O. N. Temkin, *Chemical Reaction Networks: A Graph-Theoretical Approach*, 1ST, Routledge, **1996** (p. 36).

[46] M. Badertscher, K. Bischofberger, M. E. Munk, E. Pretsch, *Journal of Chemical Information and Computer Sciences* **2001**, *41*, 889–893, DOI 10.1021/ci000135o (p. 37).

[47] M. Sun, S. Zhao, C. Gilvary, O. Elemento, J. Zhou, F. Wang, *Briefings in Bioinformatics* **2020**, *21*, 919–935, DOI 10.1093/bib/bbz042 (p. 39).

[48] D. Jiang, Z. Wu, C. Y. Hsieh, G. Chen, B. Liao, Z. Wang, C. Shen, D. Cao, J. Wu, T. Hou, *Journal of Cheminformatics* **2021**, *13*, 12:2021, DOI 10.1186/s13321-020-00479-8 (p. 39).

[49] V. Fung, J. Zhang, E. Juarez, B. G. Sumpter, *npj Computational Materials* **2021**, *7*, 84:2021, DOI 10.1038/s41524-021-00554-0 (p. 39).

[50] T. A. Young, J. J. Silcock, A. J. Sterling, F. Duarte, *Angewandte Chemie - International Edition* **2021**, *60*, 4266–4274, DOI `10.1002/anie.202011941` (p. 39).

[51] S. Pablo-García, S. Morandi, R. A. Vargas-Hernández, K. Jorner, N. López, A. Aspuru-Guzik, *ChemRxiv* **2022**, DOI `10.26434/chemrxiv-2022-m719x` (p. 39).

[52] E. Petrus, M. Segado, C. Bo, *Chemical Science* **2020**, *11*, 8448–8456, DOI `10.1039/d0sc03530k` (p. 39).

[53] R. Bader, *Atoms in Molecules — a Quantum Theory*, *Vol. 360*, Clarendon Press, **1994** (p. 39).

[54] G. Landrum, RDKit: Open-source Cheminformatics, **2006**, `http://www.rdkit.org/` (visited on Nov. 11, 2022) (p. 39, 162).

[55] N. M. O'Boyle, M. Banck, C. A. James, C. Morley, T. Vandermeersch, G. R. Hutchison, *Journal of Cheminformatics* **2011**, *3*, 33:2011, DOI `10.1186/1758-2946-3-33` (p. 39).

[56] D. Weininger, *Journal of Chemical Information and Computer Sciences* **1988**, *28*, 31–36, DOI `10.1021/ci00057a005` (p. 39).

[57] D. Weininger, A. Weininger, J. L. Weininger, *Journal of Chemical Information and Computer Sciences* **1989**, *29*, 97–101, DOI `10.1021/ci00062a008` (p. 39).

[58] M. Quirós, S. Gražulis, S. Girdzijauskaitė, A. Merkys, A. Vaitkus, *Journal of Cheminformatics* **2018**, *10*, 23, DOI `10.1186/s13321-018-0279-6` (p. 40).

[59] V. D. Hähnke, S. Kim, E. E. Bolton, *Journal of Cheminformatics* **2018**, *10*, 36, DOI `10.1186/s13321-018-0293-8` (p. 40).

[60] R. Gómez-Bombarelli, J. N. Wei, D. Duvenaud, J. M. Hernández-Lobato, B. Sánchez-Lengeling, D. Sheberla, J. Aguilera-Iparraguirre, T. D. Hirzel, R. P. Adams, A. Aspuru-Guzik, *ACS Central Science* **2018**, *4*, 268–276, DOI `10.1021/acscentsci.7b00572` (p. 40).

[61] S. Heller, A. McNaught, S. Stein, D. Tchekhovskoi, I. Pletnev, *Journal of Cheminformatics* **2013**, *5*, 7:2013, DOI 10.1186/1758-2946-5-7 (p. 40).

[62] S. Heller, *Journal of Cheminformatics* **2014**, *6*, 7:2013, DOI 10.1186/1758-2946-6-S1-P4 (p. 40).

[63] N. M. O'Boyle, A. Dalke, *ChemRxiv* **2018**, DOI 10.26434/chemrxiv.7097960.v1 (p. 40).

[64] M. Krenn, F. Häse, A. K. Nigam, P. Friederich, A. Aspuru-Guzik, *Machine Learning: Science and Technology* **2020**, *1*, 045024, DOI 10.1088/2632-2153/aba947 (p. 40).

[65] J. C. Brammer, G. Blanke, C. Kellner, A. Hoffmann, S. Herres-Pawlis, U. Schatzschneider, *Journal of Cheminformatics* **2022**, *14*, 66, DOI 10.1186/s13321-022-00640-5 (p. 40).

[66] C. R. Harris et al., *Nature* **2020**, *585*, 357–362, DOI 10.1038/s41586-020-2649-2 (p. 42).

[67] P. Virtanen et al., *Nature Methods* **2020**, *17*, 261–272, DOI 10.1038/s41592-019-0686-2 (p. 42).

[68] J. D. Hunter, *Computing in Science & Engineering* **2007**, *9*, 90–95, DOI 10.1109/MCSE.2007.55 (p. 42).

[69] A. A. Hagberg, D. A. Schult, P. J. Swart in 7th Python in Science Conference (SciPy 2008), **2008**, pp. 11–15 (p. 42, 102).

[70] T. Berners-Lee, J. Hendler, O. Lassila, *Scientific American* **2001**, *284*, 34–43, DOI 10.1038/scientificamerican0501-34 (p. 43).

[71] T. Berners-Lee, Uniform Resource Identifier (URI): Generic Syntax, **2005**, https://datatracker.ietf.org/doc/html/rfc3986 (visited on Nov. 11, 2022) (p. 43).

[72] T. Bray, J. Paoli, C. Sperberg-McQueen, E. Maier, F. Yergeau, Extensible Markup Language (XML) 1.0, Fifth Edition, **2008**, https://www.w3.org/TR/xml/ (visited on Nov. 11, 2022) (p. 43).

[73] R. Cyganiak, D. Wood, M. Lanthaler, Resource Description Framework (RDF) 1.1: Concepts and Abstract Syntax, **2014**, `http://www.w3.org/TR/rdf-concepts/` (visited on Nov. 11, 2022) (p. 44).

[74] T. R. Gruber, *Knowledge Acquisition* **1993**, *5*, 199–220, DOI `10.1006/knac.1993.1008` (p. 45).

[75] C. Bizer, T. Heath, T. Berners-Lee in *Semantic Services, Interoperability and Web Applications*, IGI Global, **2011**, pp. 205–227, DOI `10.4018/978-1-60960-593-3.ch008` (p. 45).

[76] Wikimedia, Wikidata, `https://www.wikidata.org/wiki` (visited on Sept. 5, 2022) (p. 45).

[77] A. Eibeck, M. Q. Lim, M. Kraft, *Computers and Chemical Engineering* **2019**, *131*, 106586, DOI `10.1016/j.compchemeng.2019.106586` (p. 45).

[78] W3C OWL Working Group, OWL 2 Web Ontology Language: Document Overview, **2012**, `https://www.w3.org/TR/owl2-overview/` (visited on Nov. 11, 2022) (p. 45).

[79] R. Jackson et al., *Database* **2021**, *2021*, baab069, DOI `10.1093/database/baab069` (p. 45, 46).

[80] M. Ashburner et al., *Nature Genetics* **2000**, 25–29, DOI `10.1038/75556` (p. 46).

[81] S. Carbon et al., *Nucleic Acids Research* **2021**, *49*, D325–D334, DOI `10.1093/nar/gkaa1113` (p. 46).

[82] M. Álvarez-Moreno, C. De Graaf, N. López, F. Maseras, J. M. Poblet, C. Bo, *Journal of Chemical Information and Modeling* **2015**, *55*, 95–103, DOI `10.1021/ci500593j` (p. 46, 103).

[83] C. Bo, F. Maseras, N. López, *Nature Catalysis* **2018**, *1*, 809–810, DOI `10.1038/s41929-018-0176-4` (p. 46).

[84] M. Álvarez-Moreno, ioChem-BD Source Code, **2022**, https://gitlab.com/ioChem-BD/iochem-bd (visited on Aug. 26, 2022) (p. 46).

[85] P. Murray-Rust, H. S. Rzepa, *Journal of Chemical Information and Computer Sciences* **1999**, *39*, 928–942, DOI 10.1021/ci990052b (p. 46, 135).

[86] P. Murray-Rust, H. S. Rzepa, *Journal of Chemical Information and Computer Sciences* **2001**, *41*, 1113–1123, DOI 10.1021/ci000404a (p. 46, 135).

[87] G. V. Gkoutos, P. Murray-Rust, H. S. Rzepa, M. Wright, *Journal of Chemical Information and Computer Sciences* **2001**, *41*, 1124–1130, DOI 10.1021/ci000406v (p. 46, 135).

[88] W. Phadungsukanan, M. Kraft, J. A. Townsend, P. Murray-Rust, *Journal of Cheminformatics* **2012**, *4*, 15, DOI 10.1186/1758-2946-4-15 (p. 46, 135).

[89] M. D. Wilkinson et al., *Scientific Data* **2016**, *3*, 160018, DOI 10.1038/sdata.2016.18 (p. 47).

[90] BlueObelisk, JUMBOconverters Library, **2020**, https://github.com/BlueObelisk/jumbo-converters (visited on Aug. 26, 2022) (p. 47).

[91] S. Grimme, *Journal of Chemical Theory and Computation* **2019**, *15*, 2847–2862, DOI 10.1021/acs.jctc.9b00143 (p. 50).

[92] P. Pracht, F. Bohle, S. Grimme, *Physical Chemistry Chemical Physics* **2020**, *22*, 7169–7192, DOI 10.1039/c9cp06869d (p. 50).

[93] I. Ismail, R. Chantreau Majerus, S. Habershon, *The Journal of Physical Chemistry A* **2022**, acs.jpca.2c06408, DOI 10.1021/acs.jpca.2c06408 (p. 50).

[94] S. Maeda, K. Morokuma, *Journal of Chemical Physics* **2010**, *132*, 1–5, DOI 10.1063/1.3457903 (p. 50).

[95] S. Maeda, K. Morokuma, *Journal of Chemical Theory and Computation* **2011**, *7*, 2335–2345, DOI 10.1021/ct200290m (p. 50).

[96] S. Maeda, T. Taketsugu, K. Morokuma, *Journal of Computational Chemistry* **2014**, *35*, 166–173, DOI 10.1002/jcc.23481 (p. 50).

[97] S. Maeda, Y. Harabuchi, M. Takagi, T. Taketsugu, K. Morokuma, *Chemical Record* **2016**, *16*, 2232–2248, DOI 10.1002/tcr.201600043 (p. 50, 51).

[98] K. Ohno, S. Maeda, *Chemical Physics Letters* **2004**, *384*, 277–282, DOI 10.1016/j.cplett.2003.12.030 (p. 51).

[99] K. Ohno, S. Maeda, *Journal of Physical Chemistry A* **2006**, *110*, 8933–8941, DOI 10.1021/jp0611491 (p. 51).

[100] S. Maeda, K. Ohno, *Journal of Physical Chemistry A* **2005**, *109*, 5742–5753, DOI 10.1021/jp0513162 (p. 51).

[101] S. Maeda, K. Ohno, K. Morokuma, *Physical Chemistry Chemical Physics* **2013**, *15*, 3683–3701, DOI 10.1039/c3cp44063j (p. 51).

[102] A. Laio, M. Parrinello, *Proceedings of the National Academy of Sciences of the United States of America* **2002**, *99*, 12562–12566, DOI 10.1073/pnas.202427399 (p. 51).

[103] A. Barducci, M. Bonomi, M. Parrinello, *Wiley Interdisciplinary Reviews: Computational Molecular Science* **2011**, *1*, 826–843, DOI 10.1002/wcms.31 (p. 51).

[104] G. Torrie, J. Valleau, *Journal of Computational Physics* **1977**, *23*, 187–199, DOI 10.1016/0021-9991(77)90121-8 (p. 51).

[105] J. Kästner, *Wiley Interdisciplinary Reviews: Computational Molecular Science* **2011**, *1*, 932–942, DOI 10.1002/wcms.66 (p. 51).

[106] M. R. Sørensen, A. F. Voter, *Journal of Chemical Physics* **2000**, *112*, 9599–9606, DOI 10.1063/1.481576 (p. 52).

[107] Y. Shim, N. B. Callahan, J. G. Amar, *Journal of Chemical Physics* **2013**, *138*, 094101, DOI 10.1063/1.4793218 (p. 52).

[108] D. R. Glowacki, E. Paci, D. V. Shalashilin, *Journal of Physical Chemistry B* **2009**, *113*, 16603–16611, DOI 10.1021/jp9074898 (p. 52, 55).

[109] D. R. Glowacki, E. Paci, D. V. Shalashilin, *Journal of Chemical Theory and Computation* **2011**, *7*, 1244–1252, DOI 10.1021/ct200011e (p. 52, 55).

[110] E. Vanden-Eijnden, M. Venturoli, *Journal of Chemical Physics* **2009**, *131*, 044120, DOI 10.1063/1.3180821 (p. 52).

[111] E. Martínez-Núñez, *Journal of Computational Chemistry* **2015**, *36*, 222–234, DOI 10.1002/jcc.23790 (p. 52).

[112] E. Martínez-Núñez, *Physical Chemistry Chemical Physics* **2015**, *17*, 14912–14921, DOI 10.1039/c5cp02175h (p. 52).

[113] E. Martínez-Núñez, G. L. Barnes, D. R. Glowacki, S. Kopec, D. Peláez, A. Rodríguez, R. Rodríguez-Fernández, R. J. Shannon, J. J. Stewart, P. G. Tahoces, S. A. Vazquez, *Journal of Computational Chemistry* **2021**, *42*, 2036–2048, DOI 10.1002/jcc.26734 (p. 52, 55).

[114] L.-P. Wang, A. Titov, R. McGibbon, F. Liu, V. S. Pande, T. J. Martínez, *Nature Chemistry* **2014**, *6*, 1044–1048, DOI 10.1038/nchem.2099 (p. 52).

[115] S. L. Miller, H. C. Urey, *Science* **1959**, *130*, 245–251, DOI 10.1126/science.130.3370.245 (p. 52).

[116] S. Kopec, E. Martínez-Núñez, J. Soto, D. Peláez, *International Journal of Quantum Chemistry* **2019**, *119*, 1–22, DOI 10.1002/qua.26008 (p. 55).

[117] D. T. Gillespie, *Journal of Computational Physics* **1976**, *22*, 403–434, DOI 10.1016/0021-9991(76)90041-3 (p. 55).

[118] A. Rodríguez, R. Rodríguez-Fernández, S. A. Vázquez, G. L. Barnes, J. J. P. Stewart, E. Martínez-Núñez, *Journal of Computational Chemistry* **2018**, *39*, 1922–1930, DOI 10.1002/jcc.25370 (p. 55).

[119] D. Garay-Ruiz, Amk-Tools, **2021**, `https://github.com/dgarayr/amk_tools` (visited on Sept. 13, 2022) (p. 56).

[120] D. Garay-Ruiz, M. Álvarez-Moreno, C. Bo, E. Martínez-Núñez, *ACS Physical Chemistry Au* **2022**, *2*, 225–236, DOI `10.1021/acsphyschemau.1c00051` (p. 56, 140).

[121] Bokeh Development Team, Bokeh: Python Library for Interactive Visualization, **2018**, `https://bokeh.pydata.org/en/latest/` (visited on Sept. 13, 2022) (p. 61).

[122] R. B. Van Order, H. G. Lindwall, *Chemical Reviews* **1942**, *30*, 69–96, DOI `10.1021/cr60095a004` (p. 65).

[123] E. R. Radwanski, R. L. Last, *Plant Cell* **1995**, *7*, 921–934, DOI `10.1105/tpc.7.7.921` (p. 66).

[124] J. H. Lee, J. Lee, *FEMS Microbiology Reviews* **2010**, *34*, 426–444, DOI `10.1111/j.1574-6976.2009.00204.x` (p. 66).

[125] H. Shui, Y. Zhou, H. Li, Z. Wang, Z. Lei, S. Ren, C. Pan, W. Wang, *Fuel* **2013**, *108*, 385–390, DOI `10.1016/j.fuel.2012.11.005` (p. 66).

[126] L. Zhang, M. Zhang, J. Gao, D. Xu, S. Zhou, Y. Wang, *Energy and Fuels* **2018**, *32*, 9358–9370, DOI `10.1021/acs.energyfuels.8b02297` (p. 66).

[127] J. Liu, X. Zhang, B. Hu, Q. Lu, D.-j. Liu, C.-q. Dong, Y.-p. Yang, *Journal of the Energy Institute* **2020**, *93*, 649–657, DOI `10.1016/j.joei.2019.05.015` (p. 66, 74, 76, 77).

[128] M. Corval, M.-F. Lautié, *Organic Mass Spectrometry* **1983**, *18*, 198–201, DOI `10.1002/oms.1210180504` (p. 66).

[129] A. Laskin, A. Lifshitz, *Symposium (International) on Combustion* **1998**, *27*, 313–320, DOI `10.1016/S0082-0784(98)80418-9` (p. 66).

[130] M. G. Nix, A. L. Devine, B. Cronin, M. N. Ashfold, *Physical Chemistry Chemical Physics* **2006**, *8*, 2610–2618, DOI 10.1039/b603499c (p. 67).

[131] A. L. Sobolewski, W. Domcke, C. Dedonder-Lardeux, C. Jouvet, *Physical Chemistry Chemical Physics* **2002**, *4*, 1093–1100, DOI 10.1039/b110941n (p. 67).

[132] W. M. Irvine et al., *Nature* **1996**, *383*, 418–420, DOI 10.1038/383418a0 (p. 67).

[133] R. D. Brown, *Nature* **1977**, *270*, 39–41, DOI 10.1038/270039a0 (p. 67).

[134] G. L. Blackman, R. D. Brown, P. D. Godfrey, H. I. Gunn, *Nature* **1976**, *261*, 395–396, DOI 10.1038/261395a0 (p. 67).

[135] D. Garay-Ruiz, C. Bo, ioChem-BD Data Collection: Indole Decomposition, http://dx.doi.org/10.19061/iochem-bd-1-223 (visited on Nov. 25, 2022) (p. 72).

[136] E. T. Polehampton, K. M. Menten, S. Brünken, G. Winnewisser, J. P. Baluteau, *Astronomy and Astrophysics* **2005**, *431*, 203–213, DOI 10.1051/0004-6361:20041598 (p. 81).

[137] B. A. McGuire, A. M. Burkhardt, S. Kalenskii, C. N. Shingledecker, A. J. Remijan, E. Herbst, M. C. McCarthy, *Science* **2018**, *359*, 202–205, DOI 10.1126/science.aao4890 (p. 81).

[138] M. A. Zdanovskaia, B. J. Esselman, R. C. Woods, R. J. McMahon, *Journal of Chemical Physics* **2019**, *151*, 024301, DOI 10.1063/1.5100805 (p. 81).

[139] K. L. K. Lee, B. A. McGuire, M. C. McCarthy, *Physical Chemistry Chemical Physics* **2019**, *21*, 2946–2956, DOI 10.1039/c8cp06070c (p. 81).

[140] S. Hoops, R. Gauges, C. Lee, J. Pahle, N. Simus, M. Singhal, L. Xu, P. Mendes, U. Kummer, *Bioinformatics* **2006**, *22*, 3067–3074, DOI 10.1093/bioinformatics/btl485 (p. 86, 166).

[141] R. J. Kee, F. M. Rupley, E. Meeks, J. a Miller, Chemkin-III: A Fortran Chemical Kinetics Package for the Analysis of Gas-Phase Chemical and Plasma Kinetics, tech. rep., Sandia National Laboratories (SNL), Albuquerque, NM, and Livermore, CA (United States), **1996**, pp. 3–164, DOI 10.2172/481621 (p. 86).

[142] D. R. Glowacki, C. H. Liang, C. Morley, M. J. Pilling, S. H. Robertson, *Journal of Physical Chemistry A* **2012**, *116*, 9545–9560, DOI 10.1021/jp3051033 (p. 86).

[143] T. Bligaard, J. K. Nørskov, S. Dahl, J. Matthiesen, C. H. Christensen, J. Sehested, *Journal of Catalysis* **2004**, *224*, 206–217, DOI 10.1016/j.jcat.2004.02.034 (p. 86).

[144] I. A. Filot, R. A. Van Santen, E. J. Hensen, *Angewandte Chemie - International Edition* **2014**, *53*, 12746–12750, DOI 10.1002/anie.201406521 (p. 86).

[145] D. Fantauzzi, T. Zhu, J. E. Mueller, I. A. Filot, E. J. Hensen, T. Jacob, *Catalysis Letters* **2015**, *145*, 451–457, DOI 10.1007/s10562-014-1448-5 (p. 86).

[146] C. F. Goldsmith, R. H. West, *Journal of Physical Chemistry C* **2017**, *121*, 9970–9981, DOI 10.1021/acs.jpcc.7b02133 (p. 86).

[147] A. H. Motagamwala, J. A. Dumesic, *Chemical Reviews* **2021**, *121*, 1049–1076, DOI 10.1021/acs.chemrev.0c00394 (p. 86).

[148] M. Jaraíz, L. Enríquez, R. Pinacho, J. E. Rubio, A. Lesarri, J. L. López-Pérez, *The Journal of Organic Chemistry* **2017**, *82*, 3760–3766, DOI 10.1021/acs.joc.7b00220 (p. 86).

[149] M. Kalek, F. Himo, *Journal of the American Chemical Society* **2017**, *139*, 10250–10266, DOI 10.1021/jacs.7b01931 (p. 86).

[150] L. Artús Suàrez, Z. Culakova, D. Balcells, W. H. Bernskoetter, O. Eisenstein, K. I. Goldberg, N. Hazari, M. Tilset, A. Nova, *ACS Catalysis* **2018**, *8*, 8751–8762, DOI 10.1021/acscatal.8b02184 (p. 86).

[151] M. Besora, F. Maseras, *Wiley Interdisciplinary Reviews: Computational Molecular Science* **2018**, *8*, 1–13, DOI `10.1002/wcms.1372` (p. 86, 165).

[152] R. Pérez-Soto, M. Besora, F. Maseras, *Organic Letters* **2020**, *22*, 2873–2877, DOI `10.1021/acs.orglett.0c00367` (p. 86).

[153] J. Heitkämper, J. Herrmann, M. Titze, S. M. Bauch, R. Peters, J. Kästner, *ACS Catalysis* **2022**, *12*, 1497–1507, DOI `10.1021/acscatal.1c05440` (p. 86).

[154] J. A. Dumesic, G. W. Huber, M. Boudart in *Handbook of Heterogeneous Catalysis*, (Eds.: G. Ertl, H. Knözinger, F. Schüth, J. Weitkamp), Wiley-VCH Verlag GmbH & Co. KGaA, Weinheim, Germany, **2008**, pp. 1–15, DOI `10.1002/9783527610044` (p. 87).

[155] A. P. Umpierre, E. de Jesús, J. Dupont, *ChemCatChem* **2011**, *3*, 1413–1418, DOI `10.1002/cctc.201100159` (p. 87).

[156] S. Kozuch, C. Amatore, A. Jutand, S. Shaik, *Organometallics* **2005**, *24*, 2319–2330, DOI `10.1021/om050160p` (p. 87, 109).

[157] S. Kozuch, S. Shaik, *Journal of the American Chemical Society* **2006**, *128*, 3355–3365, DOI `10.1021/ja0559146` (p. 87).

[158] S. Kozuch, S. Shaik, *Journal of Physical Chemistry A* **2008**, *112*, 6032–6041, DOI `10.1021/jp8004772` (p. 87).

[159] S. Kozuch, S. E. Lee, S. Shaik, *Organometallics* **2009**, *28*, 1303–1308, DOI `10.1021/om800772g` (p. 87, 109).

[160] S. Kozuch, J. M. Martin, *Chemical Communications* **2011**, *47*, 4935–4937, DOI `10.1039/c1cc10717h` (p. 87).

[161] S. Kozuch, J. M. Martin, *ChemPhysChem* **2011**, *12*, 1413–1418, DOI `10.1002/cphc.201100137` (p. 87).

[162] S. Kozuch, *Wiley Interdisciplinary Reviews: Computational Molecular Science* **2012**, *2*, 795–815, DOI `10.1002/wcms.1100` (p. 87, 100).

[163] S. Kozuch, *ACS Catalysis* **2015**, *5*, 5242–5255, DOI 10.1021/acscatal.5b00694 (p. 91, 96).

[164] E. Solel, N. Tarannam, S. Kozuch, *Chemical Communications* **2019**, *55*, 5306–5322, DOI 10.1039/c9cc00754g (p. 91–93).

[165] D. Garay-Ruiz, C. Bo, *ACS Catalysis* **2020**, *10*, 12627–12635, DOI 10.1021/acscatal.0c02332 (p. 91).

[166] D. Garay-Ruiz, gTOFfee, **2020**, https://gitlab.com/dgarayr/gtoffee (visited on Aug. 26, 2022) (p. 91, 102).

[167] Z. Mao, C. T. Campbell, *ACS Catalysis* **2019**, *9*, 9465–9473, DOI 10.1021/acscatal.9b02761 (p. 102, 122).

[168] R. Meir, S. Kozuch, A. Uhe, S. Shaik, *Chemistry - A European Journal* **2011**, *17*, 7623–7631, DOI 10.1002/chem.201002724 (p. 109).

[169] A. Poater, X. Solans-Monfort, E. Clot, C. Copéret, O. Eisenstein, *Journal of the American Chemical Society* **2007**, *129*, 8207–8216, DOI 10.1021/ja070625y (p. 109).

[170] S. Kozuch, J. M. Martin, *ACS Catalysis* **2011**, *1*, 246–253, DOI 10.1021/cs100129u (p. 109).

[171] M. García-Melchor, M. C. Pacheco, C. Nájera, A. Lledós, G. Ujaque, *ACS Catalysis* **2012**, *2*, 135–144, DOI 10.1021/cs200526x (p. 109).

[172] A. S. Dudnik, V. L. Weidner, A. Motta, M. Delferro, T. J. Marks, *Nature Chemistry* **2014**, *6*, 1100–1107, DOI 10.1038/nchem.2087 (p. 109).

[173] W. Wang, G.-C. Wang, *RSC Advances* **2015**, *5*, 83459–83470, DOI 10.1039/C5RA14815D (p. 109).

[174] J. González-Fabra, F. Castro-Gómez, W. M. C. Sameera, G. Nyman, A. W. Kleij, C. Bo, *Catalysis Science and Technology* **2019**, *9*, 5433–5440, DOI 10.1039/c9cy01285k (p. 109, 112).

[175] P. C. Kamer, J. N. Reek, P. W. van Leeuwen, *Rhodium Catalyzed Hydroformylation*, *Vol. 22*, (Eds.: P. W. N. M. Van Leeuwen, C. Claver), Springer Netherlands, Dordrecht, **2005**, DOI 10.1002/3527605134.ch6 (p. 110).

[176] G. G. Stanley in *Kirk-Othmer Encyclopedia of Chemical Technology*, **2017**, pp. 1–19, DOI 10.1002/0471238961.1524150209121.a01.pub2 (p. 110).

[177] D. M. Hood, R. A. Johnson, A. E. Carpenter, J. M. Younker, D. J. Vinyard, G. G. Stanley, *Science* **2020**, *367*, 542–548, DOI 10.1126/science.aaw7742 (p. 110).

[178] L. E. Rush, P. G. Pringle, J. N. Harvey, *Angewandte Chemie - International Edition* **2014**, *53*, 8672–8676, DOI 10.1002/anie.201402115 (p. 110, 113).

[179] R. L. Martin, P. J. Hay, L. R. Pratt, *Journal of Physical Chemistry A* **1998**, *102*, 3565–3573, DOI 10.1021/jp980229p (p. 112).

[180] D. H. Wertz, *Journal of the American Chemical Society* **1980**, *102*, 5316–5322, DOI 10.1021/ja00536a033 (p. 112).

[181] J. Cooper, T. Ziegler, *Inorganic Chemistry* **2002**, *41*, 6614–6622, DOI 10.1021/ic020294k (p. 112).

[182] G. Morales, R. Martínez, T. Ziegler, *Journal of Physical Chemistry A* **2008**, *112*, 3192–3200, DOI 10.1021/jp711204v (p. 112).

[183] B. Yang, R. Burch, C. Hardacre, G. Headdock, P. Hu, *ACS Catalysis* **2012**, *2*, 1027–1032, DOI 10.1021/cs2006789 (p. 119).

[184] W. Wang, Y. Wang, G. C. Wang, *Physical Chemistry Chemical Physics* **2018**, *20*, 2492–2507, DOI 10.1039/c7cp06693g (p. 119).

[185] T. T. Xiao, R. S. Li, G. C. Wang, *Journal of Physical Chemistry C* **2020**, *124*, 6611–6623, DOI 10.1021/acs.jpcc.9b11347 (p. 119).

[186] W. Q. Yan, J. B. Zhang, R. J. Zhou, Y. Q. Cao, Y. A. Zhu, J. H. Zhou, Z. J. Sui, W. Li, D. Chen, X. G. Zhou, *Industrial and Engineering Chemistry Research* **2020**, *59*, 22451–22459, DOI `10.1021/acs.iecr.0c04525` (p. 119).

[187] R. Y. Brogaard, U. Olsbye, *ACS Catalysis* **2016**, *6*, 1205–1214, DOI `10.1021/acscatal.5b01957` (p. 119).

[188] T. Salavati-Fard, S. Caratzoulas, R. F. Lobo, D. J. Doren, *ACS Catalysis* **2017**, *7*, 2240–2246, DOI `10.1021/acscatal.6b02682` (p. 119).

[189] M. A. Ortuño, V. Bernales, L. Gagliardi, C. J. Cramer, *Journal of Physical Chemistry C* **2016**, *120*, 24697–24705, DOI `10.1021/acs.jpcc.6b06381` (p. 119).

[190] X. Wang, X. Zhang, R. Pandharkar, J. Lyu, D. Ray, Y. Yang, S. Kato, J. Liu, M. C. Wasson, T. Islamoglu, Z. Li, J. T. Hupp, C. J. Cramer, L. Gagliardi, O. K. Farha, *ACS Catalysis* **2020**, *10*, 8995–9005, DOI `10.1021/acscatal.0c01844` (p. 119).

[191] W. Xue, X. Song, D. Mei, *Journal of Physical Chemistry C* **2021**, *125*, 17097–17108, DOI `10.1021/acs.jpcc.1c04906` (p. 119).

[192] Y. Luo, Z. Chen, J. Zhang, Y. Tang, Z. Xu, D. Tang, *RSC Advances* **2017**, *7*, 13473–13486, DOI `10.1039/c6ra27207j` (p. 119).

[193] M. Abdelgaid, J. Dean, G. Mpourmpakis, *Catalysis Science and Technology* **2020**, *10*, 7194–7202, DOI `10.1039/d0cy01474e` (p. 119).

[194] C. S. Praveen, A. P. Borosy, C. Copéret, A. Comas-Vives, *Inorganic Chemistry* **2021**, *60*, 6865–6874, DOI `10.1021/acs.inorgchem.0c03135` (p. 119).

[195] S. Zhang, Y. Tang, L. Nguyen, Y. F. Zhao, Z. Wu, T. W. Goh, J. J. Liu, Y. Li, T. Zhu, W. Huang, A. I. Frenkel, J. Li, F. F. Tao, *ACS Catalysis* **2018**, *8*, 110–121, DOI `10.1021/acscatal.7b01788` (p. 119).

[196] M. Busch, M. D. Wodrich, C. Corminboeuf, *Chemical Science* **2015**, *6*, 6754–6761, DOI 10.1039/c5sc02910d (p. 119).

[197] M. D. Wodrich, B. Sawatlon, M. Busch, C. Corminboeuf, *ChemCatChem* **2018**, *10*, 1586–1591, DOI 10.1002/cctc.201701709 (p. 119).

[198] M. D. Wodrich, B. Sawatlon, E. Solel, S. Kozuch, C. Corminboeuf, *ACS Catalysis* **2019**, *9*, 5716–5725, DOI 10.1021/acscatal.9b00717 (p. 119, 120).

[199] M. D. Wodrich, B. Sawatlon, M. Busch, C. Corminboeuf, *Accounts of Chemical Research* **2021**, *54*, 1107–1117, DOI 10.1021/acs.accounts.0c00857 (p. 119).

[200] P. Sabatier, *La Catalyse En Chimie Organique*, (Ed.: C. Beranger), Librairie Polytechnique, **1913**, DOI 10.14375/NP.9782369430186 (p. 119).

[201] A. Balandin in *Advances in Catalysis*, **1969**, pp. 1–210, DOI 10.1016/S0360-0564(08)60029-2 (p. 119).

[202] S. Pablo-García, R. García-Muelas, A. Sabadell-Rendón, N. López, *Wiley Interdisciplinary Reviews: Computational Molecular Science* **2021**, *11*, 1–16, DOI 10.1002/wcms.1540 (p. 119).

[203] M. Cohen, D. G. Vlachos, *Industrial & Engineering Chemistry Research* **2022**, *61*, 5117–5128, DOI 10.1021/acs.iecr.2c00390 (p. 121).

[204] M. Cohen, D. G. Vlachos, *Industrial & Engineering Chemistry Research* **2022**, acs.iecr.2c01991, DOI 10.1021/acs.iecr.2c01991 (p. 121).

[205] Y.-F. Zhao, Y. Yang, C. Mims, C. H. F. Peden, J. Li, D. Mei, *Journal of Catalysis* **2011**, *281*, 199–211, DOI 10.1016/j.jcat.2011.04.012 (p. 124).

[206] A. Uhe, S. Kozuch, S. Shaik, *Journal of Computational Chemistry* **2011**, *32*, 978–985, DOI 10.1002/jcc.21669 (p. 125).

[207] K. Degtyarenko, P. de Matos, M. Ennis, J. Hastings, M. Zbinden, A. Mcnaught, R. Alcántara, M. Darsow, M. Guedj, M. Ashburner, *Nucleic Acids Research* **2008**, *36*, 344–350, DOI 10.1093/nar/gkm791 (p. 132).

[208] J. Hastings, G. Owen, A. Dekker, M. Ennis, N. Kale, V. Muthukrishnan, S. Turner, N. Swainston, P. Mendes, C. Steinbeck, *Nucleic Acids Research* **2016**, *44*, D1214–D1219, DOI 10.1093/nar/gkv1031 (p. 132).

[209] European Molecular Biology Laboratory - European Bioinformatics Institute (EMBL-EBI), Chemical Entities of Biological Interest (ChEBI), **2004**, https://www.ebi.ac.uk/chebi/# (visited on Nov. 11, 2022) (p. 132).

[210] C. Pachl, N. Frank, J. Breitbart, S. Bräse, *ArXiv* **2020**, DOI 10.48550/arXiv.2002.03842 (p. 132).

[211] P. Strömert, J. Hunold, A. Castro, S. Neumann, O. Koepler, *Pure and Applied Chemistry* **2022**, *94*, 605–622, DOI 10.1515/pac-2021-2007 (p. 132).

[212] P. Sankar, G. Aghila, *Journal of Chemical Information and Modeling* **2006**, *46*, 2355–2368, DOI 10.1021/ci050533x (p. 132).

[213] P. Sankar, G. Aghila, *Journal of Chemical Information and Modeling* **2007**, *47*, 1747–1762, DOI 10.1021/ci700043u (p. 132).

[214] D. Vijayasarathi, P. Sankar, *Journal of Molecular Graphics and Modelling* **2015**, *61*, 30–43, DOI 10.1016/j.jmgm.2015.06.001 (p. 132).

[215] C. Batchelor, Chemical Reactions Ontology (RXNO), **2021**, https://github.com/rsc-ontologies/rxno (visited on Nov. 11, 2022) (p. 132, 139).

[216] C. Batchelor, Chemical Methods Ontology (CHMO), https://github.com/rsc-ontologies/rsc-cmo (visited on Nov. 11, 2022) (p. 133).

[217] F. Tao, Q. Qi, *Nature* **2019**, *573*, 490–491 (p. 133).

[218] J. Bai, L. Cao, S. Mosbach, J. Akroyd, A. A. Lapkin, M. Kraft, *JACS Au* **2022**, *2*, 292–309, DOI 10.1021/jacsau.1c00438 (p. 133).

[219] J. Hastings, L. Chepelev, E. Willighagen, N. Adams, C. Steinbeck, M. Dumontier, *PLoS ONE* **2011**, *6*, (Ed.: F. Fraternali), e25513, DOI 10.1371/journal.pone.0025513 (p. 133).

[220] E. L. Willighagen, J. Alvarsson, A. Andersson, M. Eklund, S. Lampa, M. Lapins, O. Spjuth, J. E. Wikberg, *Journal of Biomedical Semantics* **2011**, *2*, S6, DOI 10.1186/2041-1480-2-S1-S6 (p. 133).

[221] Chemical Semantics Inc, The Gainesville Core Ontology, 0.7.0, **2015**, http://ontologies.makolab.com/gc/gc.html (visited on Nov. 11, 2022) (p. 133).

[222] F. Farazi, J. Akroyd, S. Mosbach, P. Buerger, D. Nurkowski, M. Salamanca, M. Kraft, *Journal of Chemical Information and Modeling* **2020**, *60*, 108–120, DOI 10.1021/acs.jcim.9b00960 (p. 133).

[223] F. Farazi, N. B. Krdzavac, J. Akroyd, S. Mosbach, A. Menon, D. Nurkowski, M. Kraft, *Computers and Chemical Engineering* **2020**, *137*, 106813, DOI 10.1016/j.compchemeng.2020.106813 (p. 133, 135).

[224] N. Krdzavac, S. Mosbach, D. Nurkowski, P. Buerger, J. Akroyd, J. Martin, A. Menon, M. Kraft, *Journal of Chemical Information and Modeling* **2019**, *59*, 3154–3165, DOI 10.1021/acs.jcim.9b00227 (p. 133).

[225] B. Wang, P. A. Dobosh, S. Chalk, M. Sopek, N. S. Ostlund, *The Journal of Physical Chemistry A* **2017**, *121*, 298–307, DOI 10.1021/acs.jpca.6b10489 (p. 135).

[226] G. Fu, C. Batchelor, M. Dumontier, J. Hastings, E. Willighagen, E. Bolton, *Journal of Cheminformatics* **2015**, *7*, 1–15, DOI 10.1186/s13321-015-0084-4 (p. 139).

[227] D. Garay-Ruiz, OntoRXN, **2022**, https://gitlab.com/dgarayr/ontorxn (visited on Nov. 11, 2022) (p. 144).

[228] D. Garay-Ruiz, OntoRXN-Tools, **2022**, https://gitlab.com/dgarayr/ontorxn_tools (visited on Nov. 11, 2022) (p. 144).

[229] J. B. Lamy, *Artificial Intelligence in Medicine* **2017**, *80*, 11–28, DOI 10.1016/j.artmed.2017.07.002 (p. 145).

[230] RDFLib, RDFLib, **2022**, https://github.com/RDFLib/rdflib (visited on Nov. 11, 2022) (p. 145).

[231] S. Harris, A. Seaborne, W3C Consortium, SPARQL 1.1 Query Language, **2013**, https://www.w3.org/TR/sparql11-query/ (visited on Nov. 11, 2022) (p. 146).

[232] P. Bansal, A. Morgat, K. B. Axelsen, V. Muthukrishnan, E. Coudert, L. Aimo, N. Hyka-Nouspikel, E. Gasteiger, A. Kerhornou, T. B. Neto, M. Pozzato, M.-C. Blatter, A. Ignatchenko, N. Redaschi, A. Bridge, *Nucleic Acids Research* **2022**, *50*, D693–D700, DOI 10.1093/nar/gkab1016 (p. 147).

[233] J. Galgonek, J. Vondrášek, *Journal of Cheminformatics* **2021**, *13*, 38:2021, DOI 10.1186/s13321-021-00515-1 (p. 147).

[234] D. Garay-Ruiz, C. Bo, *Chemistry - A European Journal* **2021**, *27*, 11618–11626, DOI 10.1002/chem.202100755 (p. 148, 149, 151, 155, 156).

[235] D. Garay-Ruiz, C. Bo, ioChem-BD Data Collection: Peroxyformate Decomposition, https://doi.org/10.19061/iochem-bd-1-198 (visited on Nov. 25, 2022) (p. 148, 149).

[236] R. E. Pincock, *Journal of the American Chemical Society* **1962**, *84*, 312–313, DOI 10.1021/ja00861a041 (p. 148).

[237] R. E. Pincock, *Journal of the American Chemical Society* **1964**, *86*, 1820–1826, DOI 10.1021/ja01063a033 (p. 148).

[238] M. J. Frisch et al., Gaussian09 Revision D.01, Gaussian Inc., Wallingford CT, **2010** (p. 149, 163).

[239] J. D. Chai, M. Head-Gordon, *Physical Chemistry Chemical Physics* **2008**, *10*, 6615–6620, DOI 10.1039/b810189b (p. 149).

[240] R. Krishnan, J. S. Binkley, R. Seeger, J. A. Pople, *The Journal of Chemical Physics* **1980**, *72*, 650–654, DOI 10.1063/1.438955 (p. 149, 163).

[241] A. D. McLean, G. S. Chandler, *The Journal of Chemical Physics* **1980**, *72*, 5639–5648, DOI 10.1063/1.438980 (p. 149, 163).

[242] A. V. Marenich, C. J. Cramer, D. G. Truhlar, *The Journal of Physical Chemistry B* **2009**, *113*, 6378–6396, DOI 10.1021/jp810292n (p. 149, 163).

[243] C. Maquilón, B. Limburg, V. Laserna, D. Garay-Ruiz, J. González-Fabra, C. Bo, M. Martínez Belmonte, E. C. Escudero-Adán, A. W. Kleij, *Organometallics* **2020**, *39*, 1642–1651, DOI 10.1021/acs.organomet.9b00773 (p. 163).

[244] D. Garay-Ruiz, C. Bo, J. González-Fabra, ioChem-BD Data Collection: Regio- and Stereoisomeric Formation of Bicyclic Organic Carbonates, https://doi.org/10.19061/iochem-bd-1-141 (visited on Nov. 25, 2022) (p. 163).

[245] H. L. Schmider, A. D. Becke, *Journal of Chemical Physics* **1998**, *108*, 9624–9631, DOI 10.1063/1.476438 (p. 163).

[246] A. D. Becke, *Journal of Chemical Physics* **1997**, *107*, 8554–8560, DOI 10.1063/1.475007 (p. 163).

[247] Y. Sato, N. Hosaka, K. Yamamoto, H. Inomata, *Fluid Phase Equilibria* **2010**, *296*, 25–29, DOI 10.1016/j.fluid.2009.12.030 (p. 168, 169).

[248] S. Faranda, G. Foca, A. Marchetti, G. Pályi, L. Tassi, C. Zucchi, *Journal of Molecular Liquids* **2004**, *111*, 117–123, DOI 10.1016/j.molliq.2003.12.008 (p. 169).

[249] D. Garay-Ruiz, CRN-KG Web Service, **2022**, `https://doi.org/10.19061/crn-kg-ontorxn.2022` (visited on Nov. 11, 2022) (p. 170).

[250] Blazegraph, BlazeGraph RDF Database, **2015**, `https://github.com/blazegraph/database` (visited on Nov. 11, 2022) (p. 171).

UNIVERSITAT
ROVIRA i VIRGILI