Towards the Computer-Aided Molecular Design

of Reactants and Products

by

Vikrant Arjun Dev

A dissertation submitted to the graduate faculty of
Auburn University
in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Auburn, Alabama
December 16, 2017

Keywords: Computer-Aided Molecular Design, Machine Learning, Molecular Descriptors

Approved by

Mario R. Eden, Chair, Professor of Chemical Engineering
Nishanth G. Chemmangattuvalappil, Associate Professor of Chemical Engineering
Allan E. David, Assistant Professor of Chemical Engineering
Jin Wang, Graduate Program Chair, Professor of Chemical Engineering
Steven E. Taylor, Associate Dean of Research, Professor of Biosystems Engineering

Abstract

Many chemical processes generate products with properties of interest to businesses and end consumers (B&EC), using reactions. Thus, a need for quantitative modeling of physico-chemical properties and molecular design (MD) in reactive systems has arisen. Quantitative modeling precedes MD and is useful in relating chemical structure to properties of interest. Using quantitative models, a chemical's properties can be systematically varied by varying its structure. This structure variation, without sole reliance on intuition, assists in exploring a large portion of the chemical space. When the rising prowess of computers is tapped, such an exploration is termed as computer-aided molecular design (CAMD). CAMD of products of reactions is thus beneficial since the demands of B&EC can be met efficiently. Since products originate from reactants, CAMD of products will also lead to the CAMD of reactants. While CAMD of solvents and catalysts has received significant attention, there is a paucity of CAMD algorithms that design reactants and products. To address this paucity, CAMD of reactants and products in three scenarios has been explored in this work. In the first scenario, only the products' respective dominant properties are optimized, given a set of property constraints. In the second scenario, properties that are dependent on the structures of both reactants and products are optimized. Unlike the first scenario, both reactants and products are subject to property constraints. In the third scenario, each reactant and product's respective dominant property is optimized. Like the second scenario, both reactants and products are subject to property constraints. Our CAMD

methodologies incorporate property models with a variety of molecular descriptors using signature descriptors, which are molecular building blocks. In order to generate feasible structures, previously developed structural constraints have been improved. Since the structures of reactants and products are related, relationships have been derived between them using signature descriptors. To demonstrate the efficacy of the developed CAMD methodologies, a case study has been solved for each scenario. Additionally, for CAMD of reactants, products and solvents for reaction rate optimization, we compare promising ensemble learning algorithms' abilities to model reaction rate constant in terms of structures of reactants and solvents. We assessed decision tree-based ensemble methods' abilities to model the Diels-Alder reaction's rate constant in a case study.

Acknowledgements

Foremost, I want to express my deepest gratitude to my advisor Dr. Mario Eden and to my co-advisor Dr. Nishanth Chemmangattuvalappil. It is with their unwavering support and trust in my abilities that I am currently at this momentous juncture in my life. I am especially thankful to them for fostering independence and for encouraging me to explore fresh ideas and concepts. I would also like to thank Dr. Eden for providing me the opportunity to present our work at various conferences, both within and outside the United States of America. Directly or indirectly my stay at Auburn has been productive and comfortable because of Dr. Eden and Dr. Chemmangattuvalappil and I am thankful to them for this. I am also thankful for the support extended by my committee members Dr. Steven Taylor, Dr. Allan David and Dr. Jin Wang. Their feedback during the development of my dissertation was valuable as it helped me enhance the quality of my work.

During my stay at Auburn, I was surrounded by many friendly and intelligent students who added to the joy of studying at Auburn. From Dr. Eden's group, I would like to acknowledge the support of Dr. Zhihong Yuan, Dr. Zheng Liu, Dr. Charles Solvason, Dr. Susilpa Bommareddy, Dr. Subin Hada, Dr. Robert Herring III, Mr. Colin Haser, Dr. Narendra Sadhwani, Dr. Anjan Tula, Dr. Sawitree Kalakul, Mrs. Sarah Davis, Mr. Shounak Datta, Mr. Bernardo Lousada and Mr. Pengcheng Li. Outside of Dr. Eden's group, I would like to acknowledge the support of Dr. Rajeshwar Chinnawar, Dr. Shantanu Pradhan, Mr. Mahesh Parit, Mr. Wangyu Ma, Dr. Pranav Vengsarkar, Dr.

Venkataramesh Pallapolu, Dr. Achintya Sujan, Mr. Vignesh Venkatasubramanian, Dr. Abhijeet Phalle, Dr. Shaima Nahreen, Dr. Joyanta Goswami, Dr. Tapas Acharya, Dr. Peng Cheng, Mr. Partha Chatterjee, Dr. Wenjian Guan and Mr. Andreas Biesinger. I would also like to appreciate the support of my collaborators Dr. Lik Yin Ng and Mr. Shounak Datta.

Finally, I would like to acknowledge the love and support of my mother, Mrs. Rekha Dev, who has stood by me like a rock. Her umpteen sacrifices and her unceasing encouragement has made me who I am today. I am also deeply thankful to my uncle and aunt, Mr. Vipin Mahajan and Mrs. Krina Mahajan, and my grand uncle and grand aunt Mr. Kishan Mahajan and Mrs. Joyce Mahajan for their love, support and encouragement. I am also greatly thankful to my maternal grandmother, Mrs. Krishna Mahajan, who is now deceased, for her encouragement and her love.

Table of Contents

List of Figures

List of Tables

1. Introduction

During the period 1980-2000, the chemical industry transformed itself from being process-centered to product-centered (Stephanopoulos and Reklaitis, 2011). In the former direction, the products were simple molecules and the R&D activities were not as complicated as in the latter direction. In the product-centered approach, the chemical industry moved towards the manufacture and sale of high value-added materials marketed on performance rather than compositional specifications (Hill, 2009). These materials were termed as *chemical products*. This shift in direction is also reflected in the paper of Grossmann and Westerberg (2000). They stated that increased investor pressure in the coming years will create demands for improved earnings performance from both commodity and specialty product manufacture. They further stated that driving forces like these will lead to process design expanding to accommodate product design, with particular emphasis on design of new molecules. Such a paradigm shift in process design has appeared in accordance with the predictions alongside other adaptations in process systems engineering (PSE), of which process design is a part. PSE, in general, has diversified from its process roots, first into wider aspects of project management, then to multi-site operations, and eventually to consideration of the whole supply chain (Sargent, 2005). In line with the expansion efforts of the PSE community, Klatt and Marquardt (2009) exhorted them to collaborate with other disciplines like material sciences which actively pursue product design. They called for the widening of scope of PSE into multi-scale product and process systems engineering (MPPSE) which would address product design among other areas in an integrated manner. Adjiman and Galindo (2011) later coined the term *Molecular Systems Engineering* that formally recognizes the design of molecules and materials as an integral part of the overall task of designing and

optimizing processes and products. Integrated product development and design also featured as one of the most important PSE conference topics in the period 1985-2006, as part of emerging areas (Glavič, 2012). Thus, from the above one can safely claim that there has been an attempt to elevate the relevance of product design over the years in PSE and by extension, in the chemical engineering community.

In the international PSE conference, Foundations of Computer Aided Process Design (FOCAPD), convened in 2005 at Princeton University, the overarching theme was "Discovery through Product and Process Design" (Stephanopoulos and Reklaitis, 2011). The participants recognized that the notion of a product covered a fairly broad range of entities. These ranged from simple small molecules, functional molecules such as dyes to structured products which perform certain functions such as batteries and products closely connected to emotional disposition of humans such as clothing. Most of the efforts from the PSE community on product design, however, have been focused on the optimal generation of molecular structures, which satisfy a set of desired specifications usually expressed in terms of physico-chemical properties. Stephanopoulos and Reklaitis (2011) and Mlinar (2015) list a variety of reasons why this is the case. Some of the prime reasons are the significant enlargement in the design space with the rise in the scale of products considered and reduction in reliability of mathematical models relating structures to properties. For example, Segall (2012) questions in his paper whether we can currently carry out computer-aided drug design (CADD). He states that the prediction in drug discovery is not yet sufficient to permit a design paradigm, as demonstrated by the large number of compounds that must be synthesized and tested to identify a successful drug. However, he does not diminish the utility of computational tools but provides a future outlook where the ever-rising computational prowess

will play a significant role. Although, the end of Moore's law appears to be in sight because the size of the transistors can only be diminished to an extent, chip manufacturing companies are investing considerably in the development of post-Moore's law devices (Pavlus, 2015). These will help in significantly improving the current computational prowess. Strategies such as 'Heterotic computing' which involve usage of combination of two or more computational systems are also emerging in order to help accelerate progress in a post-Moore's law world (Kendon et al., 2015). Because of recent improvements in computer architecture and distribution techniques, Ceder (2013) suggests that we are inching towards a golden age of materials science. Although most of the efforts, over the years, concerning computer-aided product design (CAPD) have been devoted to computer-aided molecular design (CAMD), there are problems involving CAMD that are yet to be substantially addressed. For example, processes that involve reactions and high pressures need attention first in terms of developing property models (PMs) and then in developing methodologies to utilize these PMs to design molecules and processes. So far, in processes involving reactions, which is the focus of our work, CAMD of solvents, catalysts, reactants and products has been carried out. Solvents have been designed in reactive systems as mass separating agents (MSAs) and to enhance performance of reactions. In the first application of CAMD, one of the aims of MSA usage is to separate products of interest in order to drive the reaction in the forward direction. In the case of extractive fermentation, additionally, poisoning of cells by the products is to be avoided. The early approaches to design MSAs have been well documented in the work of Papadopoulos and Linke (2009). In their work, they have also presented a framework for integrated molecular and process design. In the framework, the information obtained at the solvent design stage is incorporated in the process design stage using

a data mining approach in the form of clustering. They utilized multi-objective optimization to assess various trade-offs characterizing the solvent design space. Also, Cheng and Wang (2010) presented an approach recently to design a biocompatible solvent for an extractive fermentation and separation process. Crisp and fuzzy optimization problems were formulated and solved using Mixed-Integer Hybrid Differential Evolution in their methodology. Differing from the mathematical programming based approaches generally utilized to design MSAs, an approach based on screening and experimental tests was also developed, by Faria et al. (2013). They developed a solvent selection methodology for integrated reactive-adsorptive processes as simulated moving bed reactors.

In the second application concerning solvent design, an important goal of CAMD is to capitalize on the relationship between molecular structure and parameters influencing the performance of a reaction. Strübing et al. (2010) have discussed in detail the solvent effects on reactions and the early approaches to carry out CAMD of reaction solvents. Recently, a quantum mechanical (QM) CAMD approach was proposed by Struebing et al. (2013) where the computational expense is reduced by the adoption of a surrogate model. In this QM-CAMD approach utilizing Solvation Model based on Density (SMD), the density functional theory (DFT) calculations of the rate constant are incorporated in the problem formulation. The rate constant is then solved for its optimality at 298 K. On the other hand, Zhou et al. (2015b) proposed a framework to carry out simultaneous solvent and process design due to the lack of such a framework. The Conductor-like Screening Model for Real Solvents (COSMO-RS) (Klamt, 2011) was utilized to calculate theoretical descriptors which were consequently expressed in terms of group contributions (GCs). By relating the rate constant to the descriptors, a structure-property relationship was

ultimately established. The optimization problem which included process constraints was then solved for profit maximization. This COSMO based approach has further been extended by Zhou et al. (2015a) to take into consideration multiple reactions and model uncertainties. Besides design of pure solvents, the design of mixed solvents, Gas-expanded liquids (GXLs), has also been addressed by Siougkrou et al. (2014). Also, apart from the approaches that generally utilize mathematical programming, solvent selection approaches for reaction rate enhancement involving screening have also been devised recently by Zhou et al. (2014) and Wicaksono et al. (2014).

In the second application area of CAMD which involves catalysts, Lin et al. (2005) designed transition metal catalysts by formulating optimization problems. The tabu search method was used to search the chemical space. Properties concerning the transition metal catalysts, such as density, were expressed in terms of connectivity indices. On the other hand, by using an Inverse Quantum Chemistry (Weymuth and Reiher, 2014a) approach called Gradient-Driven Molecule Construction, Weymuth and Reiher (2014b) designed Small-Molecule Fixating Catalysts. In their approach, a search using differential evolution was conducted for a ligand sphere that stabilizes a predefined central fragment. Besides the design of transition metal catalysts, the design of enzymes also holds significance. A review that addresses the recent developments, particularly in *de novo* design of enzymes, has been conducted by Świderek et al. (2015).

In the last application of CAMD in reactive systems, an important goal is to design reactants and products such that dominant properties dependent on their structures, are optimized. For the design of reactants and products, the reaction(s) occurring in the system is known prior to the

design. However, the reactants and products are structurally variable. Although much effort has been directed towards the design of solvents and catalysts in reactive systems, not much effort has been invested in the design of reactants and products. Since many chemical products are generated from reactions, to develop a comprehensive design framework, the reactions and the reactants involved should be considered (Ng et al., 2015b). In this regard, Chemmangattuvalappil and Eden (2013) developed an algorithm to generate structures of a single unknown reactant and product using signature descriptors. They related the property operators of the unknown product and unknown reactant in terms of atomic signatures of the product and reactant. The utilization of signature descriptors in the algorithm provides the advantage of being able to treat property models expressed in terms of both GCs and TIs on a single platform (Chemmangattuvalappil et al., 2010) . However, the linear relationship derived by them, between the property operators of the reactant and product, will not hold true if the property models utilized are nonlinear. Also, the formulated mathematical program in their algorithm is solved using bilevel optimization which is a computationally expensive approach. Besides the aforementioned approach, a quantum chemical approach was utilized by De Vleeschouwer et al. (2012) to design single unknown reactants and products. The discrete best-first-search (BFS) was adapted by them for the Linear Combination of Atomic Potentials (LCAP) scheme to design optimal acidic and photoacidic substituted 2-naphthols (Xiao et al., 2014). In the LCAP scheme (Wang et al., 2006), the design of molecules is conducted by searching for the optimum nuclear-electron interaction potential function that generates a molecular system with associated target properties. The external potential, which at the very minimum is the nuclear-electron interaction potential, is expressed as a linear combination of atomic potentials, hence the name of the

approach. A concern that can arise with quantum chemical methods is that the computational cost scales with the size of the molecular system (Adjiman et al., 2014; Jinich et al., 2014). However, Wang et al., (2006) have proposed a hybrid geometry/LCAP optimization scheme to reduce some of the cost. In this approach, geometry optimization is carried out after the LCAP optimization and the LCAP optimization is repeated if the property being optimized has not converged. To avoid the computational cost of fully calculating the standard free energy change of acid dissociation of substituted 2-naphthols, De Vleeschouwer et al. (2012) utilized three different approximations to represent the gas-phase acidity. It is assumed by them, during optimization, that a substituent at a particular site displays a similar contribution regardless of the substituents on other sites. If one were to extrapolate from their optimization approach, the computational burden will increase with the number of unknown reactants and products. Approximations in the calculation of free energy change of reaction can also introduce sizeable errors in such a case. Also, in the optimization scheme of De Vleeschouwer et al. (2012), it is not clear how property constraints will be taken into account. A similar inverse approach also has been utilized by them (De Vleeschouwer et al., 2015, 2013) to optimize the stability, nucleophilicity and electrophilicity of thiadiazinyl radicals.

As can be inferred from the available methodologies for CAMD of reactants and products, there is room for methodologies that efficiently design multiple unknown reactants and products. In a generalizable methodology involving CAMD of multiple reactants and products, a systematic approach will be required that takes into account the reaction mechanism involved. Since the reactants and products are structurally related, the methodology will also have to take into account these relationships between structures irrespective of their numbers. The methodology

should additionally be able to consider multiple objectives of any of the reactants and products. The methodologies available currently for design of reactants and products are lacking in these aspects. To address these shortcomings, in this work we present approaches for design of reactants and products irrespective of their numbers for three scenarios. These scenarios are described in detail in chapter 3. Property models, which can be nonlinear, involving either TIs and/or GCs have been utilized to relate various properties to structures. Signature descriptors, which are molecular building blocks, have been utilized to treat all the property models on a single platform.

Although various scenarios have been considered while developing methodologies to design reactants and products in chapter 3, the accuracy of CAMD conducted in the presented design scenarios depends on the accuracy of the property models utilized. Thus, we are limited by the predictive ability of the property models and, the quality and amount of data available to develop these models. In order for us to design reactants and products such that the rate constant of a reaction or the rate of a reaction is optimized, we require property models that relate the rate constant of a reaction with the structures of reactants and solvents utilized. Since the structures of reactants and products are related, the need to model the rate constant in terms of the structures of products, additionally, does not arise. Currently, there are not many models available that jointly capture the effect of structures of reactants and solvents on the rate constant and consequently the rate of the reaction. With respect to property models that capture the reactants' and solvent's influence, Nandi et al. (2013) developed a quantitative structure-activation barrier relationship for Diels-Alder reaction that utilizes quantum chemical descriptors. Their aim was to construct a relationship between the activation energy and the structures of the

utilized reactants and solvent. However, their data set lacked solvent variety. Recently, Datta et al. (2016) developed a quantitative structure-property relationship (QSPR) that relates the rate constant of the Diels-Alder reaction and the structures of reactants and solvent utilized. They concatenated the data set utilized by Nandi et al. (2013) and the data set obtained from the work of Zhou et al. (2014) in order to create a larger data set with slightly increased solvent diversity. The R^2 and Q^2 values of the obtained model were 0.81 and 0.86 respectively. Although the obtained model showed good performance, there is still further scope for improvement of the R^2 and Q^2 values. Additionally, the developed hybrid algorithm of Datta et al. (2016) has not been evaluated with respect to its scalability. With the aim of improving on the R^2 and Q^2 values of the model of Datta et al. (2016), in our work, we evaluated tree-based ensemble machine learning algorithms with respect to their predictive ability. Specifically, we evaluated the following algorithms:

1. Random forests
2. Regularized random forests
3. Gradient boosted regression trees
4. Extremely randomized trees.

These ensemble learners are scalable and have found wide use in many regression and classification tasks. More details on these methods can be found in section 3.4.

In order for the reader to obtain a deeper understanding and appreciation of the CAMD methodologies of reactants and products, first, in chapter 2, a theoretical background of CAMD, property modelling and concepts concerning CAMD has been provided. These concepts include

different types of molecular descriptors, including signature descriptors, which has been utilized in this work. In chapter 3, revised structural constraints and structural relationships between reactants and products are derived. Additionally, the problem formulation and the methodologies to solve the three presented design scenarios is discussed. Also, details on ensemble machine learning and tree-based ensemble learning is provided. In chapter 4, the CAMD methodologies for the three design scenarios presented in chapter 3 are exemplified using 3 case studies. Additionally, a case study involving rate constant modeling of the Diels-Alder reaction is presented. In section 5, the conclusions and future directions are provided.

Also, for the reader's reference and consideration, it is worth noting that the various sections produced in this dissertation have been published in various peer-reviewed publications. Parts of chapters 1 and 2 have been published in a co-authored book chapter entitled 'Mathematical Principles of Chemical Product Design and Strategies' (Ng et al., 2017). The book chapter appears in the book titled 'Tools For Chemical Product Design: From Consumer Products To Biomedicine', published by Elsevier. Section 3.2.2 and section 4.1 have appeared in the work of Dev et al. (2014a, 2014b). Section 3.2.1, section 3.2.3 and section 4.2 have appeared in the work of Dev et al. (2015). Section 3.2.4 and section 4.3 has appeared in the work of Dev et al. (2016). Section 3.4 and section 4.4 appear in the work of Dev et al. (2017). Parts of section 2.2 appear in the co-authored works of Datta et al. (2016a, 2016b) and Datta et al. (2017).

2. Background

2.1. Molecular Descriptors

Molecular Descriptors (MDs) are the numerical values associated with the chemical constitution for correlation of chemical structure with various physical properties, chemical reactivity or biological activity (Roy et al., 2015a). Property models (PMs) that express relationships between properties and chemical structures of molecules, utilize MDs to represent the chemical structures. Property models express a quantitative relationship between properties and structures of molecules. Hence, they are also known as Quantitative Structure-Property Relationships (QSPRs). If the property is the biological activity of a molecule, then the QSPR is known as a Quantitative Structure-Activity Relationships (QSAR). Similarly, depending on the property other variations of the term Quantitative Structure-Property Relationships can be derived.

Todeschini and Consonni (2000) provide an alternative definition for MDs as:

The molecular descriptor is the final result of a logical and mathematical procedure which transforms chemical information encoded within a symbolic representation of a molecule into a useful number or the result of some standardized experiment.

They are thus bifurcating MDs into two types; theoretical descriptors and experimental measurements. Theoretical descriptors are numerical values that are obtained from symbolic representation of molecules while experimental measurements are values of physico-chemical properties like polarizability and dipole moment. Theoretical descriptors offer an advantage over

experimental measurements in that the error due to experimental noise can be avoided. Also, the practice of expressing the PMs in terms of other physicochemical properties is an older one. These properties themselves are now expressed in terms of theoretical descriptors. A wide variety of theoretical descriptors have been developed in terms of which different properties can be expressed. There is no consensus, however, for a set of rules or criteria that guide the development of new theoretical descriptors for various property models (Hong et al., 2012). However, some general criteria have been suggested (Roy et al., 2015b):

1. A descriptor must be correlated with the structural features and show negligible correlation with other descriptors.
2. A descriptor should be applicable to a broad class of compounds.
3. A descriptor can be calculated rapidly and does not depend on experimental properties; it can be considered more suitable than one that is computationally exhaustive and relies heavily on experimental results.
4. A descriptor should generate dissimilar values for structurally different molecules, even if the structural differences are small. This means that the descriptor should show minimal degeneracy. In addition to degeneracy, a descriptor should be continuous. It signifies that small structural changes should lead to small changes in the value of the descriptor.
5. It is always important that the descriptor has some form of physical interpretability to encode the query features of the studied molecules.
6. Another significant aspect is the ability to map descriptor values back to the structure for visualization purposes. These visualizations are sensible only when descriptor values can be associated to structural features.

Besides the classification used by Todeschini and Consonni (2000) there are also other types of classification of molecular descriptors. For example, descriptors can be classified based on origin. Based on origin, MDs can be classified as topological (graph theory based), constitutional (functional group count), geometrical (distances, valence angles, surfaces, etc.), quantum-chemical (charge distribution related), and thermodynamic (heat of formation, entropy, etc.) descriptors (Hong et al., 2012). However, the majority of researchers that develop PMs prefer to classify MDs based on their dimensionality (Roy et al., 2015b). The MDs can be classified as zero-dimensional (0D), one dimensional (1D), two dimensional (2D), etc. MDs up to seven dimensions (7D) have been developed so far. It is worth noting that descriptors up to two dimensions are the most commonly utilized ones. However, when large molecules are involved, descriptors with more than 2 dimensions are also utilized in property models. Descriptors with more than 3 dimensions are geared for more sophisticated applications and hence are not commonly used. A brief overview of descriptors of different dimensions has been provided as follows. Also, special attention has been provided to signature descriptors which are 2D descriptors since we will be utilizing them during CAMD of reactants and products.

2.1.1. 0D Descriptor

Molecular descriptors that are derived from the molecular formula fall in the category of 0D descriptors. Since while writing the molecular formula we are not concerned with the arrangement of molecules but only the composition, the descriptors are derived from a zero-dimensional representation of the molecule. Thus, the descriptors are referred to as 0D descriptors. Examples of 0D descriptors include atom counts, charge, molecular weight, etc.

2.1.2. 1D Descriptor

If fragments (e.g. substructural fragments or functional groups) of a molecule are used for molecular representation then 1D descriptors are obtained. This is because only one dimension is required to depict the type of substitution or fragments present. 1D descriptors can serve to quickly scan the chemical space for candidates based on some established similarity criteria with respect to a reference molecule. These have been used to filter out structures in the early stages of drug design. An example of such a descriptor is fragment count.

2.1.3. 2D Descriptor

2D descriptors are derived from 2D representation of molecules that takes into account the types of atoms, their number and their connection pattern with each other. Examples of 2D descriptors include chiral center count, which provides the number of chiral centers and rotatable bonds count, which provide the number of bonds capable of rotation (Roy et al., 2015a). The descriptors derived from the graphical representation of molecules are categorized under 2D descriptors. In the graphical representation, the molecule is referred to as a molecular graph. A molecular graph, G, consists of atoms which form the vertices of the graph and the covalent bonds which form the edges in the graph. Thus, atoms that have at least one bond between them are connected by an edge. The various fragments that can be obtained from G can be represented as sub-graphs. The sub-graphs thus consist of subset of edges belonging to the edge set, E, and subset of vertices belonging to the vertex set, V. The descriptors obtained from the graphical representation are termed as topological indices (TIs). These are the most widely used descriptors in model development and hence in computer-aided molecular design. TIs are very convenient to use

because they can be easily computed. Because isomorphic graphs have identical values for a given TI, TIs are graph invariants i.e. their values are independent of labelling of the molecular graph. In the following subsections, some details on the most widely used TIs both in modelling and CAMD algorithms are being provided.

2.1.3.1. Connectivity Indices

The connectivity index (CI) was introduced by Randic (1975) and since then has been modified into different forms. The connectivity index is usually denoted by the symbol χ. One usually finds 2 superscripts and one subscript assigned to the CI (Sabljic et al., 1990). The superscript on the left is a non-negative number and denotes the order of the CI and the superscript on the right, *v* denotes if a valence delta value has been utilized for calculation. The CIs are divided into 4 sub-classes: path (denoted by subscript *p*), cluster (denoted by subscript *c*), path/cluster (denoted by subscript *pc*) and chain (denoted by subscript *ch*). These subclasses are describing the substructural units considered while calculating the CIs. For example, the path based CI is calculated using paths. A path is a sequence of edges from one vertex to another end vertex, such that the edges don't repeat while traversing this sequence of edges. In most cases the subscript *p* is removed and path type is considered as a default. CIs are usually calculated from hydrogen suppressed graphs. In such molecular graphs, we do not draw/consider the hydrogen atoms. Consider the example of the mth order valence connectivity index ${}^{m}\chi_k{}^{v}$. It is defined as follows (Mu and He, 2011):

$$ {}^{m}\chi_k^{v} = \sum_{j=1}^{n_m} \left(\prod_{i=1}^{m+1} \delta_i^{v} \right)_j^{-0.5} \qquad (2.1) $$

$$\delta_i^v = (\mathrm{Z}_i^v - H_i) / (\mathrm{Z}_i - \mathrm{Z}_i^v - 1) \quad (2.2)$$

$$\delta_i = (\mathrm{Z}_i^v - H_i) \quad (2.3)$$

Where, k denotes a contiguous path type fragment, which is divided into paths (*p)*, clusters (*c*), paths/clusters (*pc*) and chains (*ch*). n_m is the number of relevant path type fragments. δ_i^v is the valence delta value calculated as shown in Eq. (2.2). In Eq. (2.2), Z_i^v is the number of valence electrons, H_i is the number of hydrogen atoms connected to atom *i*, Z_i is the number of electrons of atom *i*. If we calculate the m^{th} order connectivity index ${}^m\chi_k$, then δ_i will be substituted instead of δ_i^v in Eq. (2.1) to obtain ${}^m\chi_k$. δ_i is the degree of the atom *i* obtained from the hydrogen supressed graph. Hence H_i is subtracted from Z_i^v in Eq. (2.3). Consider the 3-methyl hexane molecule shown in Fig. 2.1. The degree values, δ_i, of each of the atoms have also been displayed.

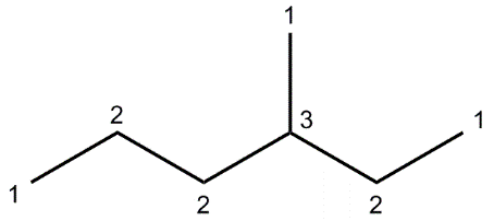

Figure 2.1: 3-methyl hexane molecule

The ${}^1\chi$ value of the 3-methyl hexane molecule can be calculated as:

${}^1\chi = (1\cdot2)^{-0.5}+(2\cdot2)^{-0.5}+(2\cdot3)^{-0.5}+(3\cdot1)^{-0.5}+(3\cdot2)^{-0.5}+(2\cdot1)^{-0.5} = 3.3081$

2.1.3.2. Functional Groups

Functional groups are widely used in group contribution models (GCMs). Functional groups are essentially molecular fragments and hence are categorized under fragment based descriptors. The groups utilized in a GCM can be of three types: first order, second order and third order. The first order consists of a large set of simple groups that allows describing the molecular structures of a wide variety of organic compounds. However, these groups capture only partially the proximity effects and are unable to distinguish among isomers. To address this drawback, higher order groups are introduced in a GCM. The second order groups permit a better description of proximity effects and differentiation among isomers. The second order groups are intended to deal with poly-functional, polar or non-polar compounds of medium size, i.e. number of carbon atoms = 3–6, and aromatic or cyclo-aliphatic compounds with only one ring and several substituents (van Speybroeck et al., 2010). The third order groups provide more structural information about molecular fragments of compounds whose description is insufficient through the first and second order groups. The third order groups account for complex heterocyclic and large (number of carbon atoms = 7–60) poly-functional acyclic compounds. According to rules developed by Marrero and Gani (2001), first-order groups should describe the entire molecule. It is also required that no atom of the molecule being considered can be included in more than one group i.e. no first order group is allowed to overlap any other first order group. Overlap between groups is however permitted for second and third order groups.

2.1.3.3. Atomic Signature Descriptors

If *G* is a molecular graph and *x* is an atom of *G*, the atomic signature of height *h* of *x* is a canonical representation of the subgraph of *G* containing all atoms that are at a distance *h* from *x* (Faulon et al., 2003; Visco et al., 2002). The signature is represented as a tree in graphical form. The atoms whose signature is drawn forms the root of the tree. Thus, a signature consists of a root atom, *x*, and atoms in each n^{th} out-neighborhood of *x*, where *n* varies from 1 to *h*. The atoms one path length away from the root atom form the 1^{st} out-neighborhood. The 1^{st} out-neighborhood is also the 1^{st} level of the signature tree. Similarly, one can define other *n* levels. The size of each atom's first out-neighborhood is the out-degree. It can be used as a coloring function to distinguish various atom types in a molecule. Atomic signatures can be used as building blocks to form molecules. As previously mentioned, signature descriptors in CAMD enable utilization of quantitative structure property/activity relationships (QSARs/QSPRs) employing TIs and GCMs on a single platform (Chemmangattuvalappil et al., 2010). Thus, a wide variety of property targets can be tracked. Faulon et al. (2003b) identified the relationship between topological indices (TIs), which constitute QSARs/QSPRs, and signatures. If *k* is a constant, ${}^{h}\alpha_G$ is the vector of occurrences of atomic signature of height *h* and *TI(root (*${}^{h}\Sigma$*))* is the vector of TI values calculated for each root of atomic signature:

$$TI(G) = k[{}^{h}\alpha_G \cdot TI(root({}^{h}\Sigma))] \tag{2.4}$$

It is worth noting that while a distinction is being made between GCMs and QSARs/QSPRs, GCMs can actually be considered as a special class of QSARs/QSPRs (van Speybroeck et al., 2010). The

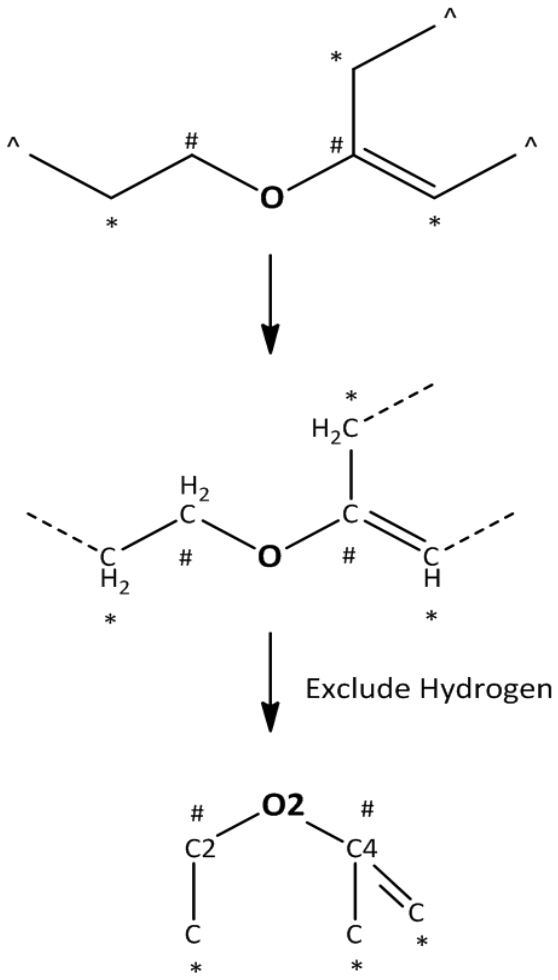

Figure 2.2: Generation of height 2 signature of O atom

groups in GCMs can be considered as fragment based descriptors (Baskin, 2008) which fall under TIs. The distinction is being made to stress that we can treat a variety of property models on a single platform using signatures in CAMD schemes. Generally, CAMD methodologies involve separate approaches for GCMs and other QSARs/QSPRs. With regards to CAMD, signature descriptors have found a place in a variety of applications which include the design of foam blowing agents (Weis et al., 2005), polymers (Brown et al., 2006), solvents (Weis and Visco, 2010),

shrinkage-reducing compounds (Kayello et al., 2014; Shlonimskaya et al., 2014) and mixtures in an integrated biorefinery (Ng et al., 2015a). Fig. 2.2 displays how a height 2 signature of the oxygen atom, shown in bold, of the shown ether molecule, is generated. The atoms marked '#' are at height 1. The atoms marked '*' are at height 2 from the root atom O. The string representation of the atomic signature of O atom is O2(C2(C)C4(=CC)). The string representation contains the first and second out-neighborhoods besides the root O atom. Each out-neighborhood is separated from other out-neighborhoods using brackets. All atoms except those at the last height are assigned colors.

2.1.4. 3D Descriptors

3D Descriptors are also known as geometrical descriptors. These descriptors are calculated by representing the molecule in the 3D space. Generally, geometrical descriptors are calculated either by utilizing optimized molecular geometry obtained by computational chemistry methods or from crystallographic coordinates (Cronin et al., 2003). 3D descriptors obtained by utilizing geometric distances between atoms constitute a special subset known as topographic indices. Now, the geometrical representation captures the relative positions of the atoms in 3D space. Thus, geometrical descriptors usually offer more information and discrimination power for similar molecular structures and molecule conformations than topological descriptors. This power to discriminate, however, comes at a cost. Since 3D descriptors require geometry optimization, a high computational expense is incurred. Additionally, complications arise if several conformations of the molecules exist. Also, it may happen that the active conformation

of the chemicals being studied for biological applications are unknown. Another concern with 3D descriptors is that, there isn't a common understanding of the necessary degree of detail to which molecular structure has to be known calculate 3D descriptors reliably (Hechinger et al., 2012). The computational methods utilized for calculation can be anywhere from molecular mechanics to ab initio methods which are more rigorous. Due to the aforementioned reasons, simpler descriptors like TIs are usually preferred for the screening of large databases of molecules and CAMD applications. On the other hand, searching for relationships between molecular structures and complex properties, such as biological activities, often require the use of 3D descriptors.

2.1.5. 4D-7D Descriptors

These descriptors have been utilized the least for CAMD applications as the computational cost incurred is even higher than 3D descriptors. These descriptors consider a variety of factors as dimensions. These include the orientation and the solvation function (Roy et al., 2015a). Such descriptors are beneficial in capturing the ligand and receptor interactions.

Although the descriptors mentioned so far have been categorized within whole number dimensions, there are descriptors that do not fit these categorizations. For example, between 2D and 3D descriptors, 2.5D descriptors exist as intermediates that tend to incorporate some aspects of the geometrical information contained in a 3D structure that were ignored by a 2D description of the molecule (Doucet and Panaye, 2010).

2.2. QSPR Modeling

QSPR modeling establishes a mathematical formalism between the behavior of a chemical and a set of quantitative chemical attributes which may be extracted from the chemical structures

using suitable theoretical and/or experimental means (Roy et al., 2015a). These attributes are also called descriptors or features of the chemical structures being studied. The naming of the study sometimes depends upon the nature of the behavior (also known as 'endpoint' and 'response') being modeled giving rise to three major classes namely quantitative structure property/activity/toxicity relationship (QSPR/QSAR/QSTR) studies taking into consideration the modeling of physicochemical properties, biological activity, and toxicological data, respectively. This nomenclature can be extended to other response variables such as cytotoxicity, reactivity, etc. However, the name 'QSPR' can be utilized to designate all such models as any type of activity based model and physicochemical based model may be considered to model the property of a given chemical. QSPR modeling, in short, entails developing a mapping between the structural features of a group of compounds and a desired property.

The efforts to develop QSPRs can be traced to as far back as the 1850s (Roy et al., 2015c). Borodin (1858) was the first to realize that a toxicological property is closely related to the chemical makeup of compounds. A similar type of behavior on the organisms was observed to be elicited by chemicals possessing same elements or taking part in similar chemical reaction. Later, Cros (1863) observed a relationship between aqueous solubility and toxicity of primary alcohols. Thus, it can be observed that the early application of QSPR modeling is associated with the field of toxicology. Around this period, Mendeleev observed a relationship among elements using their atomic weight and developed the periodic table of elements (Roy et al., 2015c). The use of atomic weight to develop the "rule of eight" by Mendeleev can be thought of as one of the oldest approaches involving "parameter" utilization in a relationship study involving chemistry.

The first proposition for the existence of a mathematical relationship between chemical structure and activity was put forward by Brown and Fraser (1868) by showing physiological activity (φ) as a mathematical function of chemical constitution (C).

$$\emptyset = f(C) \tag{2.4}$$

It needs to be mentioned that the term chemical constitution merely represented elemental composition at that period of time. This is because the concept of molecular structure was not established at that time. They showed that a series of strychnine derivatives possessing muscle paralytic activity similar to curare can be prepared by varying the quaternary substituent (Roy et al., 2015c). The modern evolution of QSPR is indebted, in part, to Corwin Hansch. Hansch et al. (1962) provided momentum to QSPR research by using Hammett constants and hydrophobicity parameters to develop correlation models on plant growth regulators. Later, the famous linear Hansch equation was developed by Hansch and Fujita (1964) by combining hydrophobic constant terms with the Hammett sigma (σ), presented as follows:

$$\log\left(\frac{1}{C}\right) = k_1\pi + k_2\sigma + k_3E_s + k\ 4 \tag{2.4}$$

Where,

k_1 , k_2, and k_3 = coefficient terms

k_4 = a constant

π = a relative hydrophobicity measure

E_s = a steric parameter

Another landmark in the historical timeline of QSPR development is the introduction of molecular connectivity index by Kier et al. (1975a). It was shown to have strong correlations to physicochemical properties (Hall et al., 1975) as well as biological activities (Kier et al., 1975b). This led to a class of new molecular descriptors and led the way for a variety of techniques aimed at differentiating molecular structures through mathematical invariants in addition the previously used physico-chemical properties.

From the aforementioned description, it can be observed that the origins of QSPR development can be traced to efforts to correlate activity of chemicals with the chemical composition which gradually led to the exploration of chemistry of compounds (Roy et al., 2015c). This trend has continued into present day approaches that have immensely benefitted from substantive developments in various computer centric research areas which are often overlapping. These include data mining, pattern recognition, machine learning, statistics, statistical learning, artificial intelligence and molecular modeling.

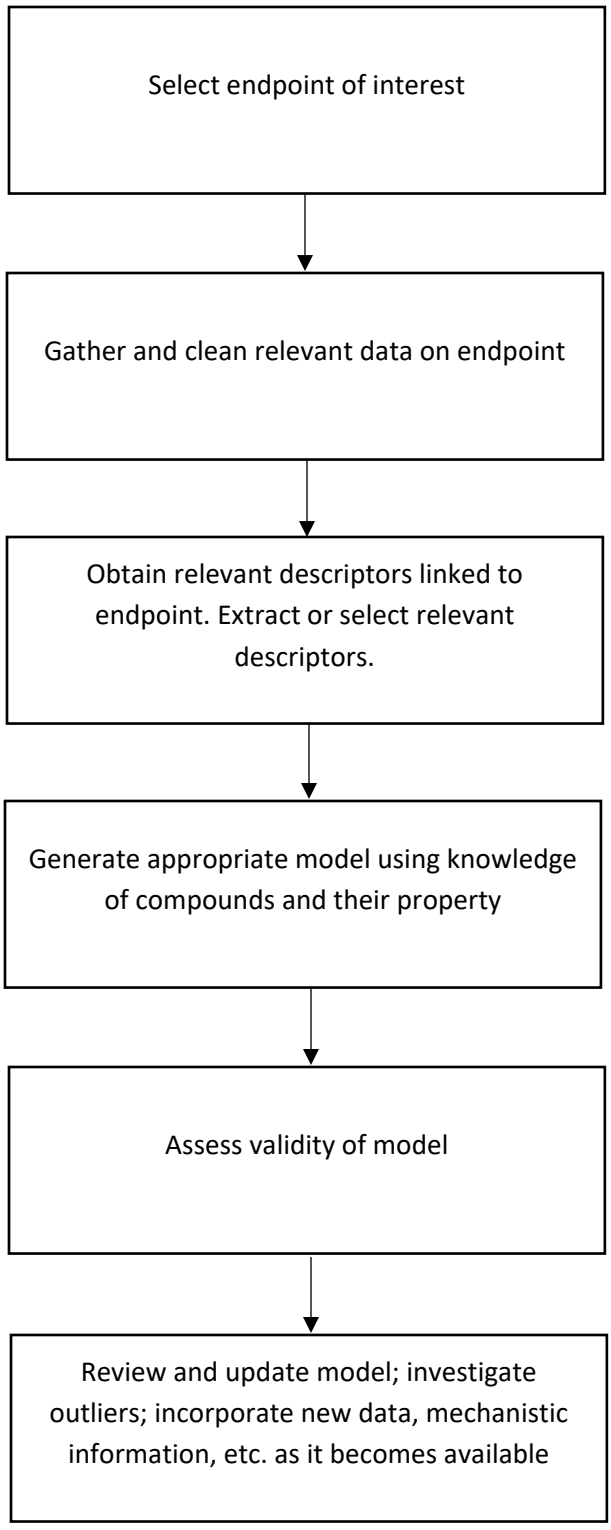

Figure 2.3: Model development and assessment workflow

Depending on the size of the data set involved and the number of descriptors chosen for model development, the process of QSPR development can involve many complex and time-consuming steps. For very large data sets, much of the effort and time is spent in the cleaning and preprocessing of the data set. Cleaning of data for large datasets may be required, for example, to deal with missing values. A workflow for generating QSPR models has been displayed in Fig. 2.3. In general, irrespective of the complexity of the steps involved, one can divide the QSPR development into the following phases:

1) Data gathering and data cleaning

2) Calculating molecular descriptors and ascertaining most informative descriptors

3) Training model using descriptor values present in the training set

4) Model validation and testing using descriptor values present in the validation and/or test set

2.2.1. Feature Selection

During the process of building property models, one is often confronted with the situation where the number of descriptors used for model building exceeds the number of data points in the training set. The data matrix then consists of more columns than rows. Here, each column corresponds to a descriptor and its values and each row corresponds to a data point in the training set. Such datasets are known as high-dimensional datasets. One of the problems of such high-dimensional datasets is that, in many cases, not all the descriptors are important for capturing the underlying phenomena of interest (Kononenko and Kukar, 2007). Varmuza and

Filzmoser (2009) provide reasons for why training the model on all descriptors' values for regression problems may not be suitable. They provide the following arguments:

1. Using all descriptors will produce a better fit of the model for the training data because the residuals become smaller. This increases the R^2 measure. However, we are usually not interested in maximizing the fit for the training data but in maximizing the prediction performance for the test data. Thus, a reduction in the number of descriptors can avoid the effects of overfitting and lead to an improved prediction performance. Overfitting is the phenomena where the model follows the errors or noise too closely (James et al., 2013a). Alternatively, when a given method yields a small training mean squared error (MSE) but a large test MSE, we are said to be overfitting the data.
2. A regression model with a high (e.g., hundreds) number of descriptors is practically impossible to interpret.
3. Using a smaller number of descriptors can also reduce the computation time considerably.

In concert with the first argument, James et al. (2013) state that, in general, adding additional features that are truly associated with the response will improve the fitted model, in the sense of leading to a reduction in test set error. However, adding noise features that are not truly associated with the response will lead to a deterioration in the fitted model, and consequently an increased test set error. This is because noise features increase the dimensionality of the problem, exacerbating the risk of overfitting without any potential upside in terms of improved test set error. This is due to noise features being assigned nonzero coefficients because of chance

associations with the response on the training set. While certain computationally expensive novel methods can construct predictive models with high accuracy from high-dimensional data, it is still of interest in most applications to reduce the dimensionality of the original data prior to any modeling (Kononenko and Kukar, 2007). Feature selection becomes necessary when there are too many attributes or the set of attributes consists of irrelevant, random, redundant, or correlated attributes that may degrade the learning performance.

Feature selection methods can be classified into the following three categories (Murty and Devi, 2015):

1. Filter methods
2. Wrapper methods
3. Embedded methods

The filter methods compute a score for each feature. Later, features are selected in accordance with the score. The filter methodology relies on the general characteristics of the training data. In this methodology, the features are selected as a preprocessing step, independent of the utilized algorithm (Bolón-Canedo et al., 2015). The filter methods are advantageous due to the low computational cost of feature selection and the good generalization ability afforded by them. Filter methods are often used in the first step of dimensionality reduction. As a first step, filters are used to remove descriptors based on mutual correlation. One approach involves retaining the descriptor that has the highest correlation with the response. The other one is relinquished. Another strategy that is utilized is the removal of descriptors having the lowest variance and the lowest correlation to the response, and retaining those descriptors that have the highest

correlation (Goodarzi et al., 2012). Filters can be divided into several categories, such as distance methods (e.g., those that utilize the Euclidean distance measure), information methods (e.g., entropy and information gain), dependency methods (e.g., correlation coefficient), and consistency methods (e.g., min-features bias). There are still many other approaches that use, for instance, mutual information, the Chi-square (χ^2) metric, the Kolmogorov-Smirnov statistic, the unbalanced correlation score, and the Shannon entropy, to select features (Goodarzi et al., 2012).

In a study by Venkatraman et al. (2004), the use of information-theoretic approaches based on the concept of mutual information gain has been applied to identify an optimal subset of descriptors for further correlation with a given biological activity. Since mutual information is a nonlinear statistical criterion, it can measure the interdependence of random variables without relying on established assumptions about their underlying relationships. This approach relies on two heuristic criteria during feature selection, namely:

1. Feature should be comparatively informative about the output
2. Feature should not be strongly dependent on other features selected

The measure of mutual information between two random variables *A* and *B* represents the amount of information about *A* contained in *B* and vice versa. When the random variables are independent of each other, the mutual information, defined in Eq. (2.5), is zero.

$$I(A,B) = \sum_{a,b} P(a,b) log \frac{P(a,b)}{P(a)P(b)} \tag{2.5}$$

The marginal probabilities for the two features are represented by $P(a)$ and $P(b)$, while $P(a,b)$ gives the joint probability. Mutual information measures the distance between the joint probability and the joint probability under the assumption of independence, $P(a)P(b)$. This technique is most suitable to problems where both descriptors and activities are categorical. In such a case where the continuous numerical variables are utilized, discretization schemes must be applied to approximate the variables.

In contrast to the filter methods, the wrapper methodology, in general, consists of utilizing the prediction performance of a given learning machine to assess the relative usefulness of subsets of variables (Guyon and Elisseeff, 2003). The wrapper approach is a more advanced approach as compared to the filter methodology. Wrapper methods often achieve better results than filters. However, they tend to be much slower than filters (Kononenko and Kukar, 2007). This is because they must repeatedly call the machine learning algorithm being utilized. Usually a greedy approach is used for searching the feature space. If a set of attributes is relatively small, more advanced methods can be used.

One technique that stands out prominently is the Genetic Algorithm (Siedlecki and Sklansky, 1989). It is especially useful for sampling large descriptor spaces. The genetic algorithm mimics the process of natural evolution whereby a population is guided towards more fitness through operations of mutation and crossover (Leardi, 2001). The fitness is measured by the error of the model that is generated. Each member of the population is represented by a chromosome. Each position of the chromosome, a gene, usually corresponds to the absence or presence of a specific variable through the binary notation. Individual chromosomes with an increased measure of

fitness are selected for the conventional operations of crossover and mutation. The fitness is typically measured by the prediction capabilities of the model resulting from the descriptors represented by the chromosome. In the process of mutation, the binary variables alter within the chromosome to either a 0 or 1, the opposite of their initial state. The process of crossover involves the selection of two chromosomes which are sliced and recombined at one or more points. The number of points of alteration depend on the type of crossover utilized. The success of a genetic algorithm is, however, owed to the careful tuning of several probability parameters. This ensures that the solution space can be effectively explored. Also, early convergence to a homogenous population which is one of the local minima, is avoided.

The genetic algorithm occupies the class known as stochastic programming/optimization. In this class, several successful techniques have been developed for the solution of problems with huge, multivariate solution spaces. Another similar technique, as that of the genetic algorithm, for variable selection is that of simulated annealing (Kirkpatrick et al., 1983). Like the genetic algorithm, it has enjoyed great success in QSPR development (Sutter and Jurs, 1995) (Itskowitz and Tropsha, 2005). Just as in the genetic algorithm's approach, simulated annealing aims to minimize the error of the generated model by iteratively modifying the subset of selected descriptors. In this process, some percentage of descriptors are exchanged for others and this new subset is tested for its ability to model the desired response variable. A probability distribution function, a Boltzmann distribution, drives the decision making of utilization of the newly selected subset of descriptors. In this process of subset selection, sometimes, the current worse solution can replace a better one. This allows the simulated annealing method to escape from the local minima of the error function. The capabilities of the simulated annealing method

stem from the temperature term in the Boltzmann distribution function which needs to be suitably altered. At an early point in the algorithm, the temperature may be higher to allow the solution to escape the trap of a local minima. As the algorithm proceeds, the temperature is reduced so that the acceptance of worse solutions becomes less probable. This often results in the identification of very high-quality solutions to the problem at hand.

While the two previously mentioned approaches of genetic algorithm and simulated annealing are stochastic in nature, there are several deterministic approaches which can explore the descriptor space more comprehensively. Most frequently, in this class, forward search and backward search are used. Kononenko and Kukar (2007) provide a brief overview of these methods in their book. Forward search starts with an empty set of attributes. In each subsequent step, it proceeds by adding either a random attribute (faster) or an attribute that optimizes some criterion (slower), and accepts the addition if the new feature subset improves the optimization criterion. Backward search starts with the complete set of attributes. In each subsequent step, it proceeds by removing either a random attribute (faster) or an attribute that optimizes some criterion (slower), and accepts the addition if the new feature subset improves the optimization criterion. Based on the two aforementioned approaches is the method called mixed search. Mixed search starts with a random set of attributes. In each subsequent step a new attribute can be added (as in forward search), or an existing attribute can be removed from the set of attributes (as in backward search). In all cases the process is repeated until the optimization criterion cannot be further improved.

In contrast to filter and wrapper methods, embedded methods do not separate the process of learning from the feature selection process. A commonly utilized embedded technique for

regression tasks is LASSO (least absolute shrinkage and selection operator) regression. LASSO carries out feature selection by shrinking some of the coefficients of the descriptors, during linear regression, to zero (Tibshirani, 1996). In the Lagrangian form, the LASSO problem is expressed as follows:

$$minimize\left\{\frac{1}{2N}\sum_{i=1}^{N}(y_i-\sum_{j=1}^{p}x_{ij}\beta_j)^2+\lambda\sum_{j=1}^{p}|\beta_j|\right\} \quad (2.6)$$

Where, *N* is the number of descriptor-response pairs (x_i, y_i). *i* varies from 1 to *N*. *p* is the number of descriptors. *j* varies from 1 to *p*. λ is a tuning parameter. βj are the coefficients. When we perform the LASSO, we are trying to find the set of coefficient estimates that lead to the smallest residual sum of squares (RSS), subject to the constraint that there is a limit *s* for how large $\sum|\beta_j|$ can be. When *s* is extremely large, then this constraint is not very restrictive, and so the coefficient estimates can be large (James et al., 2013a). When the least squares estimates have excessively high variance, the LASSO solution can yield a reduction in variance at the expense of a small increment in bias, and consequently can provide predictions with higher accuracy. Although the LASSO is widely used in variable selection, it has several drawbacks. Zou and Hastie (2005) stated that:

1. if $p > n$, the LASSO selects at most *n* variables before it saturates

2. if there is a group of variables which has very high correlation, then the LASSO tends to select only one variable from this group

3. for the usual $n > p$ condition, if there are high correlations between predictors, the prediction performance of the LASSO is dominated by ridge regression.

Zou and Hastie (2005) introduced the Elastic net method which combines beneficial features of the L1-norm and L2-norm penalties. The Elastic net is a regularized regression method which overcomes the limitations of the LASSO. This method is very useful when p is much greater than n or there are many correlated variables. The advantages are (Härdle and Simar, 2015):

1. a group of correlated variables can be selected without arbitrary omissions,

2. the number of selected variables is no longer limited by the sample size.

As an improvement on LASSO, Datta et al. (2017) recently developed the corrLASSO (correlation based adaptive least absolute shrinkage and selection operator) regression approach. corrLASSO checks for response-descriptor correlation to determine shrinkage of coefficients of descriptors and their eventual selection or removal. Using the corrLASSO, Datta et al. (2017) generated a QSPR for predicting the DNA drug binding affinity of 9-Anilinoacridine derivatives. In their developed methodology, the CorrLASSO in combination with genetic algorithm helped generate a model with superior prediction as compared with the combination of genetic algorithm and LASSO, and genetic algorithm-multiple linear regression (GA-MLR).

While feature selection methods have been divided into 3 categories, it is worth noting that hybrid methods also exist. For example, hybrid of filter and wrapper methods exist in which, first, a filter is used to generate a ranked list of features (Guyon and Elisseeff, 2006). Based on the

defined order, nested subsets of features are generated and computed using machine learning, i.e. the wrapper methodology is utilized.

2.2.2. Feature Extraction

Feature extraction is the process of determining the features to be utilized for learning. Here, we extract new features that are either linear or nonlinear combinations of the given descriptors and the extracted features are used in pattern recognition (Murty and Devi, 2015). Feature extraction is thus different from feature selection, because, the process of feature selection does not involve generating combinations of descriptors. In the context of regression and property modeling, however, feature extraction generally involves the reduction of number of variables obtained after the process of combining descriptors. Hence, in this respect, both feature selection and feature extraction can be categorized as dimensionality reduction methods.

One of the most commonly used feature extraction methods is the principal component analysis (PCA). PCA is a multivariate technique with the central aim of reducing the dimensionality of a multivariate data set while accounting for as much of the original variation as possible present in the data set. This aim is achieved by transforming to a new set of variables, the principal components, that are linear combinations of the original variables, which are uncorrelated and are ordered so that the first few of them account for most of the variation in all the original variables. PCA can thus be thought of projecting data from a higher dimensional space onto a lower dimensional space. Thus, when the data set used is highly dimensional and very noisy with a small number of samples, PCA is an appropriate method for dimensionality reduction. After the dimensionality reduction process, the regression model can be developed with the new latent

variables by implementing principal component regression (PCR). PCA is very popular within the machine learning community and is useful for visualization of data also.

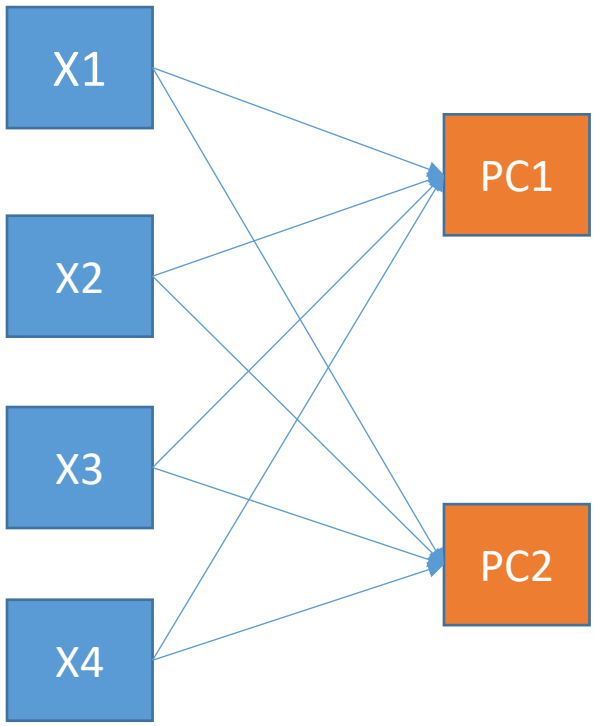

Figure 2.4: Two principal components generated from 4 descriptors

The process of PCA begins by constructing a linear projection in which each of the projected dimensions is a linear combination of the original dimensions (Rogers and Girolami, 2017). This way the most relevant information is summarized (Wold et al., 1996) (MacGregor and Kourti, 1995). Prior to implementing PCA, the data often needs to be preprocessed through a variety of techniques such that it becomes more suitable for further analysis. It is a common practice to initially mean-center and scale the values of descriptors. This is visually represented in Figure 2.5 (Eriksson et al., 2006).

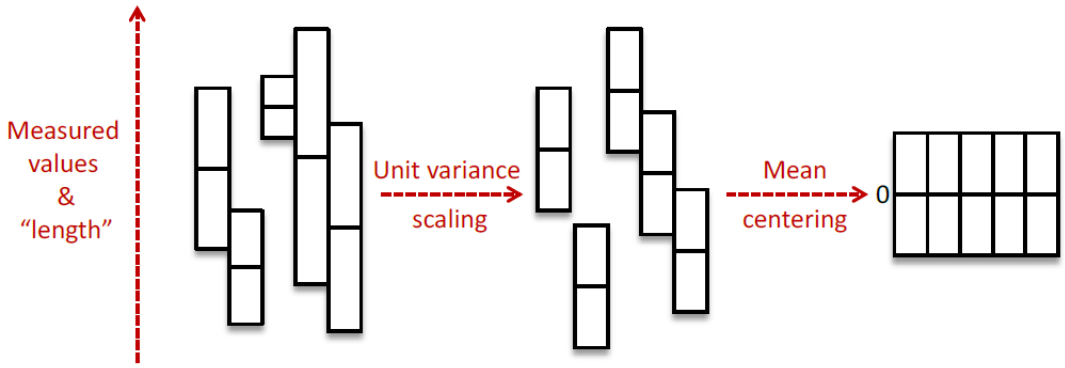

Figure 2.5: Mean centering and scaling prior to PCA

This technique ensures that no variable can dominate, in its interpreted importance, over another because of an increased length (difference in highest and lowest values) or mean value. Once the data has been preprocessed for further analysis, the process of PCA then calculates a set of principal components (PCs) by transforming the original, correlated, descriptors into a new set of uncorrelated ones. The first PC is the linear combination of the standardized original descriptors that have the greatest possible variance. Each subsequent PC is a linear combination of the standardized original variables that have the greatest possible variance, while being orthogonal to and having zero correlation with all previously defined PCs. This orthogonality constraint ensures that each variance-based axis is independent. Typically, the first three PCs capture most of the variance seen in the original data set (around 80-90%). Fig. 2.6 helps visualize the dimensionality reduction achieved through PCA.

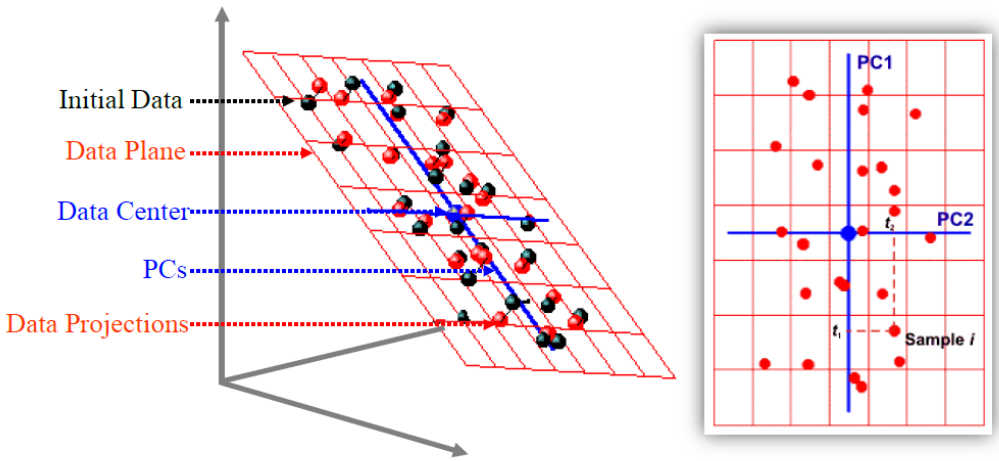

Figure 2.6: Dimensionality reduction using PCA

The loading matrix contains the coefficients in the linear combination of the original variables defining the PCs. This can be mathematically represented as shown in Eq. (2.7).

$$\boldsymbol{X}_{M\times K} = t_1 \cdot p_1^T + t_2 \cdot p_2^T + \cdots \boldsymbol{K} = \sum_{i=1}^{K} t_i \cdot p_i^T = \boldsymbol{T}_{M\times K} \cdot \boldsymbol{P}_{K\times K}^T \tag{2.7}$$

Where, **T** = the score matrix with mutually orthonormal columns

P = the loading matrix with mutually orthonormal columns

PLS is a regression extension of principal component analysis and it generalizes and combines different features from both PCA and multiple linear regressions (MLR). In addition to relating the two data matrices, of descriptors and response variables, PLS also models the common structure between them which often provides better results than those obtained with the traditional multiple regression approach. Figure 2.7 provides a visualization of the PLS process

whereby two “PCA-like” models are created for both the descriptor and response information which are then connected through an inner relationship to provide the PLS model.

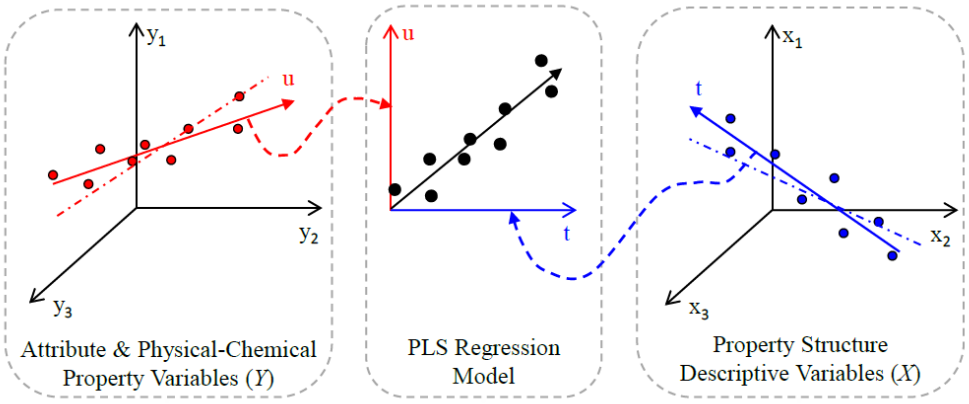

Figure 2.7: PLS regression

The score plot in Figure 2.7 shows a linear relationship between predictors (x) and responses (y), however, there may be non-linearities. The dashed-dot line seen in the outer pictures of Figure 2.7 represents the projection if PCA were performed on X and Y individually.

Both techniques of PCR and PLS aim to avoid collinearity problems which would allow one to work with several variables that is greater than the number of samples. A comparison of the two techniques (Wentzell and Vega Montoto, 2003) has revealed similar prediction capabilities, however, PCR tends to yield higher precision (degree of closeness of the measured values to each other) while PLS yields higher accuracy (degree of closeness of a measured value to the actual value).

2.2.3. Regression for QSPR Development

As discussed earlier, QSPRs are mathematical models that relate features derived from structures of compounds to their physicochemical properties or biological activities. They can be used to predict properties/activities or class (e.g., inhibitor versus noninhibitors) of compounds (Yee and Wei, 2012). In this work, however, we are concerned only with the prediction of property values, specifically, those that can aid in the design of reactants and products in reactive systems. Properties/activities are continuous variables. Although we can classify compounds based on their property/activity values into categories, we are not interested in classification of molecules in this work. Thus, here, we are focusing on regression methods and validation of models derived from regression techniques. In a regression scheme, the response variable is modeled as a function of the molecular descriptors. On the other hand, in a classification scheme, the resulting model is defined by a decision boundary, which separates the various classes within the descriptor space. The variety of available regression methods can be categorized based on whether a linear or non-linear relationship is created. Linear models are usually sufficient for creating property relationships for a dataset of similar compounds. They have the benefit of being much easier to develop and interpret when compared to other methods.

2.2.3.1. Multiple Linear Regression

Multiple linear regression (MLR) is one of the most fundamental and common modeling methods for regression QSAR. This approach models the predicted response, *Y*, by means of a set of descriptor variables, *X*, through the relationship shown in Eq. (2.6).

$$Y_{M\times L} = X_{M\times K} \cdot \beta_{K\times L} + E_{M\times L} \tag{2.6}$$

where, *M* = the number of rows of sample readings of observations

L = the number of columns of measured response properties

K = the number of columns of descriptor variables

β = the regression coefficients or sensitivities matrix

E = the error or residual matrix

There have been three cases, as described by Geladi and Kowalski (1986), for the solution of β:

1. *K* > *M*: There is no unique solution for *β* as infinite numbers of solutions exist, unless on deletes predictor variables.
2. *K* = *M*: There is one unique solution provided that *X* has full rank.
   E = *Y* - *X*·*β* = 0
3. *K* < *M*: There is no exact solution for *β*, however, a solution can be achieved by minimizing the residual in the following equation:
   E = *Y* - *X*·*β*

The most popular technique, known as the ordinary least-square (OLS) method, identifies the regression coefficients by maximizing the model sum of squares and minimizing the residual sum of squares. Using this approach, *β* can be estimated by:

$$\hat{\beta} = (X^T \cdot \mathrm{X})^{-1} \cdot \mathrm{X}^T \cdot \mathrm{Y} \tag{2.7}$$

Where, the superscript *T* symbolizes the transpose of a matrix.

When the number of *X*-variables, or descriptors, is large compared to the number of observations, this can lead to a singular (X^TX) matrix whose inverse does not exist. This happens when the number of unknown variables is greater than the number of equations, leading to an underdetermined equation system which has an infinite number of solutions for β. One solution to this problem would be to apply feature selection techniques. In addition, multivariate projection methods like PCA (principal component analysis) and PLS (partial least squares) can also be utilized to overcome such a difficulty. To date, MLR remains in use with enhancements or in combination with feature selection to improve its performance (Yee and Wei, 2012).

2.2.3.2. Nonlinear Regression

The least squares approach, as in the case of linear models, can be used to fit nonlinear models. In such a case, it is termed as nonlinear least squares. The nonlinear least squares estimate is obtained by minimizing the same objective function (loss function) as in linear regression. For the least squares estimate, the loss function is given as follows:

$$L(\beta) = \sum_{i=1}^{n} \{y_i - h(\beta, x_i)\}^2 \tag{2.8}$$

Where, *h* is a nonlinear function, β is the parameter vector and (x_i, y_i) are the training samples. If the nonlinear function, *h*, is a polynomial then the regression problem is a polynomial regression problem. The minimization of the loss function yields the values of the parameters, which ultimately determines the model. In general, when nonlinear functions are involved, there is a concern that the derived nonlinear model may overfit the data. Hence, it is necessary that

the model be tested on a test set. Usually, with regards to regression models, both linear and nonlinear, the errors are assumed to be normally distributed. However, this may not be the case when the data is explored. The data can be transformed if the normality assumptions are not satisfied. Other techniques are also available that can deal with data that does not abide by the normality assumptions.

2.2.3.2.1. Support Vector Regression

A robust method that incorporates nonlinearity is the support vector regression (SVR) method. Robustness, here, refers to the fact that the method has a high tolerance to noise. In contrast to

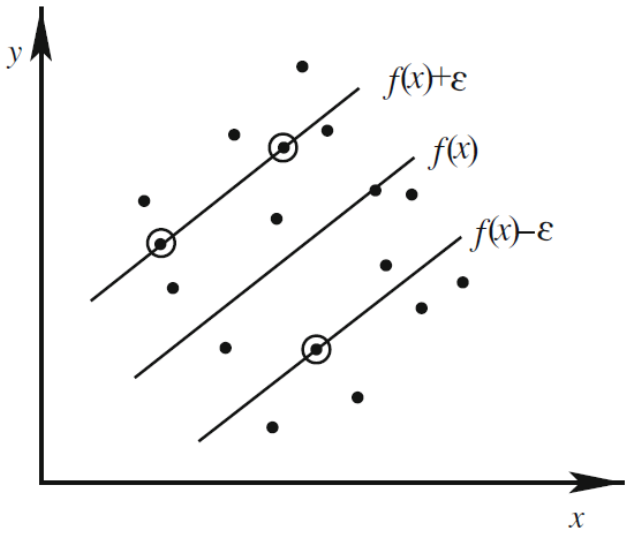

Figure 2.8: Hyperplanes in feature space of SVR. y ∉ (f(x)- ε, f(x)+ ε) is penalized.

typical regression methods, the predicted values are penalized only if their absolute error exceeds a certain user-specified threshold. Support vector regression is an extension of the support vector algorithm which was originally developed for classification tasks (Smola and

Schölkopf, 2004; Luts et al., 2010). The support vector algorithm constructs a hyperplane, or set of hyperplanes, in a highly dimensional space such that the distance to the nearest training data point is maximized (Cortes and Vapnik, 1995). The training examples that are closest to the hyperplane are called support vectors. The basic idea of SVR is to map the data into a higher-dimensional feature space via nonlinear mapping *F* and then to perform linear regression in this space (Du and Swamy, 2014). The mapping of the original finite-dimensional space into a much higher-dimensional space is made possible using kernel functions. These functions lower the computational load associated with traversing between the two mapped spaces by ensuring that dot products are easily computed in terms of the original variable space. The objective function for the SVR is the regularized risk minimization function *R(C)*.

$$R(C) = \frac{1}{2} \| w \|^2 + C \sum_{i=1}^{n} \| y_i - f(x_i) \|_\epsilon{}^2 \tag{2.9}$$

Where, $\|y_i - f(x_i)\|_\varepsilon$, is an ε-insensitive loss function used to define empirical risk as follows,

$$\| y_i - f(x_i) \|_\varepsilon = max\{0, \| y_i - f(x_i) \| - \varepsilon\} \tag{2.10}$$

and $\varepsilon > 0$ and $C > 0$ are prespecified constants. Also, *C* is a regularization constant.

In SVR, other robust statistics based loss functions such as Huber's function can also be incorporated. If a data point x_i lies inside the insensitive zone called the ε-tube, i.e., $|y_i - f(x_i)| \subseteq \varepsilon$, then it will not incur any loss. This is illustrated in Fig. 2.8 (Du and Swamy, 2014). The performance of SVR is sensitive to the hyperparameters. Thus, the hyperparameters need to be chosen suitably because they impact the model predictive ability. Not selecting appropriate

parameters may lead to underfitting or overfitting. In contrast to earlier discussed methods of PCA/PCR, PLS and OLS, SVR is built in most cases in a high-dimensional feature space (expanded feature space). SVR model is thus very flexible (Li et al., 2009). This is illustrated in Fig. 2.9 (Liang et al., 2011). With respect to application of SVR for regression modeling, Chen and Visco Jr. (2017) very recently developed an in silico pipeline for faster drug discovery. In this pipeline, they utilized SVR to predict the activity (IC_{50}) of drug candidates.

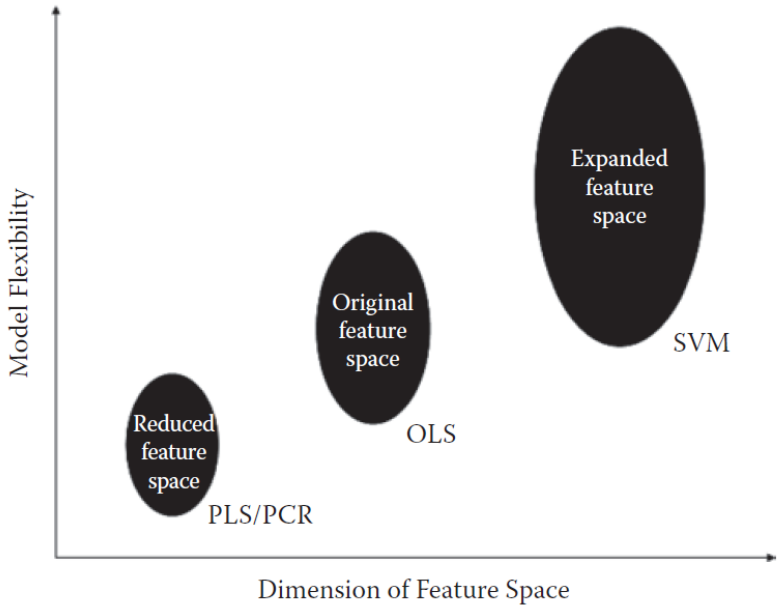

Figure 2.9: Model flexibility versus the dimension of feature space

2.2.3.2.2. Artificial Neural Networks

Artificial neural networks (ANNs) or neural nets enjoy considerable popularity with respect to development of QSPRs. Recently, learning by ANN in the form of deep learning has received even

mainstream popularity (Ekins, 2016). The development of ANN was an early attempt within the computer science community to create software that approximated the way collections of neurons in a human body function. However, it is known now that ANNs are a vastly oversimplified rendering of neural activity (Berk, 2016). However, from a modeling perspective they have been able to perform reliably. ANNs can be thought of as approximating a complicated function f(X) by a composition of many simple functional units. ANNs are generally used to combine inputs in a nonlinear fashion to arrive at an output(s). A sample neural network with an input layer, a hidden layer and an output layer has been illustrated in Fig. 2.10 (Efron and Hastie, 2016).

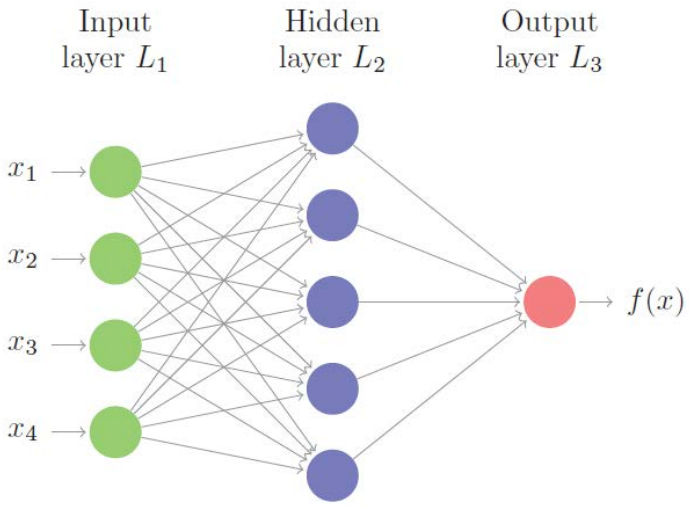

Figure 2.10: A neural network with an input layer, a hidden layer and an output layer

In general, ANNs can have multiple hidden layers. If the ANN has more than one hidden layer, then it is known as a deep neural network. The impact of the hidden layer on input variables in a simple neural network is visualized in Fig. 2.11 (Berk, 2016).

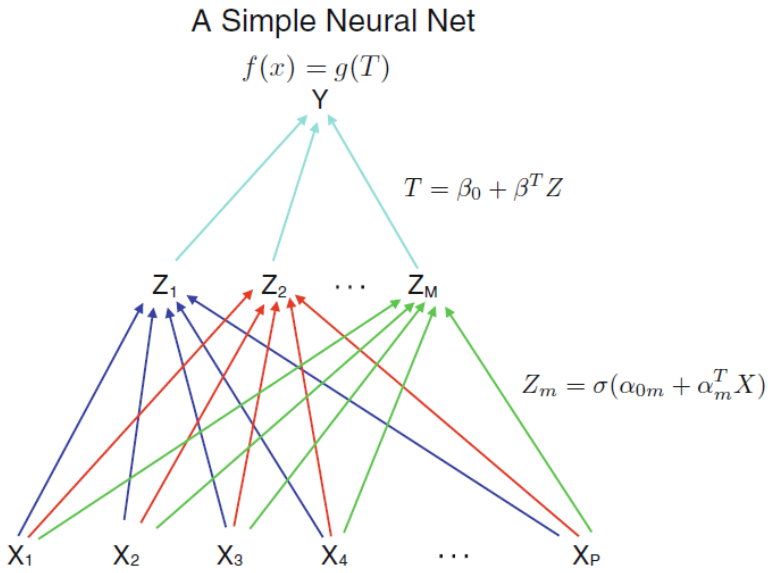

Figure 2.11: A neural net with one response, one hidden layer, and no feedback

In the illustrated neural network in Fig. 2.11, the inputs are represented by input variables x_1, x_2, ..., x_p. In QSPR development problems, these variables represent the molecular descriptors. There is a single response, *Y*, for the regression case. For other cases, e.g. for the case of classification, more complicated networks having several different outputs can be constructed. Fig. 2.11 also displays a single "hidden layer" that consists of neurons having output variables z_1, z_2, ..., z_M. These output variables associated with the hidden layer neurons can be thought of as a set of *M* unobserved, latent variables. All three components (i.e., inputs, output, and latent variables) are linked by associations that would be causal if one were trying to represent the

actions of a collection of neurons (with no feedback). In this association, each latent variable is a function of its own linear combination of the predictors. For the m^{th} latent variable, one obtains

$$Z_m = \sigma(\alpha_{0m} + \alpha_m^T X) \tag{2.11}$$

Where αs are coefficients (or weights), different for each of the M latent variables. X is the set of p inputs, and σ is the activation function, commonly a sigmoid activation function. The sigmoid function has an S-shape. The effect of the sigmoid function is that a linear combination of inputs will likely trigger an impulse such that the combination increases in value, such that the alteration towards the middle of its range alters Z_m the most. Next, a linear combination of the latent variable values is constructed as follows:

$$T = \beta_0 + \beta^T Z \tag{2.12}$$

Where the βs are the coefficients (or weights) and Z is the set of latent variables. Finally, the linear combination can be subject to the following transformation:

$$f(X) = g(T) \tag{2.13}$$

where g is the transformation function. Usually, the weights of the model are determined using the backpropagation approach. To summarize the approach, an input is fed into the network to calculate the error with respect to the desired target. This error is then used to compute the weight corrections layer by layer, backward through the net (Roy et al., 2015d). Hence, the name backpropagation. The process is repeated until the errors for the entire training set are minimized. This can involve thousands of iterations. The process of training a neural network can

thus be very time consuming. Another disadvantage that ANNs have is that they are sensitive to the data utilized. Thus, error due to variance is a concern for ANNs (Kuhn, 2016). Despite the challenges of utilizing ANNs, they have found usage in prediction of toxicological, pharmacological, and physicochemical properties, such as aquatic toxicity, drug clearance, pKa, melting point, and solubility (Mitchell, 2014). It has even found application in process modelling (Potočnik et al., 2003).

2.2.3.2.3. Regression Trees

Regression trees are decision trees that deal with a continuous response. Decision trees (Quinlan, 1986), are a type of non-linear mapping technique available for the development of QSPRs. This model that employs a divide and conquer strategy, consists of a tree-like structure containing the conventional nodes and links. In a decision tree, nodes form a hierarchical pattern, with several child nodes stemming from a common parent node. A node with no children is referred to as a leaf. Each node typically refers to a specific descriptor. At each node, a decision is made based on which the algorithm is directed towards a specific child node. This process continues till the algorithm is directed to the leaves of the tree. The final decision is based on the property class associated with that leaf. The diagram in Figure 2.12 represents the classification of a compound, based on three descriptors, d_1, d_2 and d_3, as being either active or inactive. Here, the response variable is categorical.

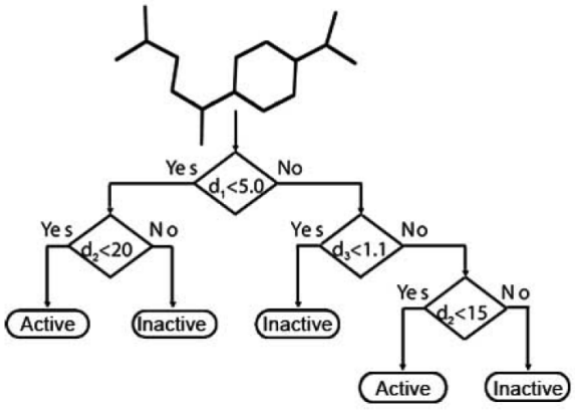

Figure 2.12: Classifying into “Active” and “Inactive” categories

One can notice that at each node, there is a split into one of two decisions based on the value of the descriptor at the node. Finally, when the leaves are reached, the final value of the response is obtained.

Regression can be thought of as classification with continuous classes. Hence, regression trees can be considered as decision trees that categorize inputs into continuous classes (Cichosz, 2015). The representation of regression tree nodes and branches is the same as for decision tree nodes and branches. In regression trees, nodes correspond to regions in the domain of the response that are divided into smaller regions. This division is achieved by splitting at each of the nodes. As is the case for decision trees, in regression trees, branches connect to descending nodes or leaves. These nodes or leaves correspond to outcomes of decision making at each split. The choice of the best split at each node of the tree is usually guided by a least squares error criterion (Rokach, 2016). Regression trees are distinct from many regression algorithms because of their connection to logic-based systems and expert systems.

In order to construct regression trees, many techniques have been developed by researchers. Of the available techniques, one of the oldest and the most utilized is the classification and regression tree (CART) methodology developed by Breiman et al. (1984). For the task of regression, the methodology analyzes the entire training set, *S*, where it explores each distinct value of each descriptor. It then locates the descriptor and its split value. The end point of methodology is that it partitions the data into two groups, S_1 and S_2, such that the overall sums of squares error is minimized. It is defined as follows:

$$SSE = \sum_{i \in S_1} (y_i - \overline{y_1})^2 + \sum_{i \in S_2} (y_i - \overline{y_2})^2 \tag{2.14}$$

Where, $\bar{y}_1$ and $\bar{y}_2$ are the averages of the training set outcomes within groups S_1 and S_2, respectively. Next, within each group, S_1 and S_2, the CART methodology locates the descriptor and its split value that best reduces SSE. Due to the application of recursive splitting during the training of regression trees, regression trees are also denoted as recursive partitioning (Kuhn and Johnson, 2013). The CART methodology has been tested in a study (Svetnik et al., 2005) on a wide range of targets, including COX-2 inhibition, blood-brain barrier permeability, CDK-2 antagonist activity, dopamine binding affinity, logD and toxicity. While they were outperformed by support vector machines and ensembles of decision trees, they did often perform better than PLS of naive bayes classifier. Ensemble machine learning using decision trees has been discussed in detail in section 3.4.

Decision trees, because of the following reasons listed by Rokach (2016) and James et al. (2013b), have become a popular machine learning method:

1. Trees are very easy to explain and understand.

2. It is believed, by some, that decision trees more closely mirror human decision-making.

3. Trees can be displayed graphically, and are easily interpreted even by a non-expert.

4. Trees can easily handle qualitative predictors without the need to create dummy variables.

5. Compared to other methods, they scale well to big data.

6. They are capable of processing datasets that may have errors or missing values.

7. They have a high predictive performance for a relatively small computational effort.

8. They are available in many open source data mining packages over a variety of platforms.

9. They are useful for various tasks like classification, regression, clustering and feature selection.

10. While decision tree algorithms might have several controlling parameters, in many cases the default values yield sufficiently good predictive performance. If tuning is still required, this usually takes a short tuning session.

While decision tree algorithms offer many advantages, they also have some drawbacks. These drawbacks have been listed by Rokach (2016). They are as follows:

1. If many complex interactions among the descriptors are present, then they do not offer good predictions.

2. They are myopic in nature because the splitting criteria is concerned only with immediate descendant descriptors.

3. They can be non-robust i.e. they suffer from high variance. A small change in the data set can cause large a large change in the estimated response. Decision trees are not suitable for noisy data.

For regression tasks, the last point can be attributed to the fact that decision trees produce piecewise-constant regression models. Regression trees divide the domain into several regions and the model's prediction attains a constant value in each region. This is visualized in Fig. 2.13 (Cichosz, 2015) for prediction of an arbitrary response *f* using *a1* and *a2* as the descriptors.

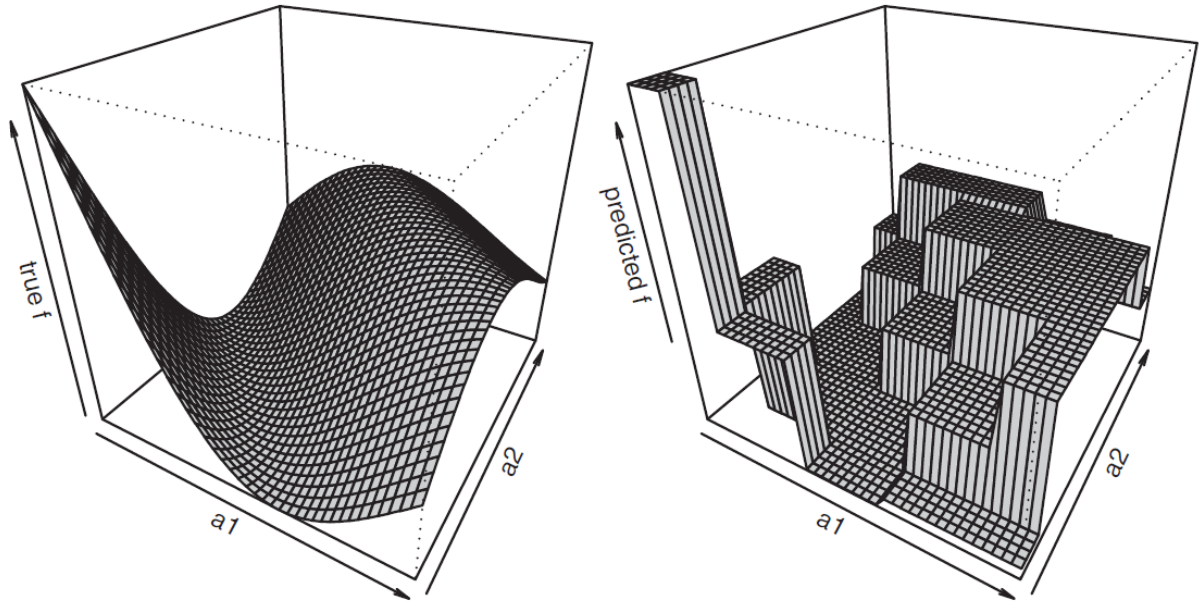

Figure 2.13: Piecewise prediction of a regression tree

While the non-robustness of decision trees is a disadvantage, ensemble methods (discussed in section 3.4) exploit this characteristic to create models that tend to have extremely good performance.

2.3. Computer-Aided Molecular Design

Experimentally, molecular design can be achieved by studying the properties of different molecules that are synthesized and then selecting the one that matches the objective the best. In such an endeavor, constraints can also be placed on the molecules that are to be synthesized, in accordance with the demands of the businesses/end consumers. The objective and constraints can be either an economic one (e.g. objective of minimization of cost), one related to the process (e.g. process yield) or the property of a molecule (e.g. boiling point). However, due to constraints on time and resources, it is infeasible to simply rely on experiments to design molecules. Also, if conflicting multiple design objectives are involved during the process of molecular design, it will be tedious to take into account all of these while designing molecules experimentally. This problem will further escalate if we want to carry out integrated product and process design where the requirement of resources can be magnanimous, if experimentally investigated. Recently, (Reymond, 2015) also enumerated all possible synthesizable and stable molecules containing up to 17 atoms. His latest database, GDB-17, contains 166.4 billion molecules. The molecules only consist of H, C, N, O, S and halogens as the atoms. One can thus understand that the chemical universe of all possible molecules consisting of all possible elements is much vaster. Thus, if we are interested in discovery of novel molecules, it will be an uphill task to scan the chemical space experimentally. It is also worth noting that much of the experimental investigations are based on intuition and/or heuristics which can be limiting factors while scanning for molecular candidates. Instead, one can harness a systematic mathematical approach that exploits the relationship between structures of molecules and their properties, and the prowess of computers. This approach, although may not completely replace experimental

research, but can at the very least aid in significantly narrowing down the molecular candidates that need to be experimentally investigated.

At its most fundamental level, computer-aided molecular design (CAMD) is the application of computer-implemented algorithms that are utilized to design a molecule for a particular application (Visco, 2010). In the context of Process Systems Engineering (PSE), CAMD plays a significant role. It helps in the design of both business-to-business (B2B) and business-to-consumer (B2C) products (Gani and Ng, 2015). An example of B2B product is vinyl chloride from which polyvinyl chloride polymer is derived. An example of B2C product is cisplatin which is a molecule known for its anti-tumor activity. By scanning the chemical space, CAMD helps in the selection of molecular candidates with attributes of interest, usually expressed in terms of physico-chemical properties.

2.3.1. CAMD Solution Strategies

Depending on the computational resources available and the time constraint of the project, different CAMD methodologies are adopted by molecule designers. The different CAMD approaches and their pros and cons are discussed in some detail in the following subsections. It is worth noting that although we have used three classifications, there exist methodologies that cannot be clearly classified into these three categories as they have characteristics borrowed from more than one category.

2.3.1.1. Virtual Screening/Database search

Searching through a database can be advantageous when the computational resources available are ample and accuracy is desired, but time is a constraint. Screening candidates can involve calculation of desired properties of each of the candidates in the database and matching it against a set of prerequisites. It can also involve similarity search where the molecules are identified based on structural similarity to some reference structure(s). In the 1960s, chemical structures were first stored as computer files in searchable form by Chemical Abstract Services, thus providing a basis for structure retrieval and searching (Willett, 1987). During the 1970s, methods for two-dimensional substructure (Cramer et al., 1974) and three-dimensional pharmacophore searching (Gund, 1977) were developed, which made it possible to search compound databases for desired structural motifs or active molecules.

If the database is huge and first principles calculations are used, one can rely on distributed computing which can help borrow idle time on different computers. The Harvard Clean Energy Project is an example of a research initiative that relies on distributed computing. The project aims to design organic photovoltaics. Utilization of supercomputers has traditionally been the route to achieve high computational resources for search of large databases. The availability of ample resources in a search involving large databases can offer advantages in terms of having the ability to use very accurate quantum chemical methods. Usually, reduction in computational expense comes with some compromise on accuracy of models. Searching of small databases may not offer as much advantage as mathematical programming based approaches because the search space is limited to molecules in the database.

2.3.1.2. Generate and Test

The Generate and Test method to solve CAMD problems was first developed for solvent selection by Gani and Brignole (1983). They were designed by keeping in mind, the functional groups that form the GCMs. The Generate and Test CAMD methodology can be divided in two stages (Cismondi and Brignole, 2004). The two stages are:

1. The generate or the molecular synthesis stage
2. The test stage or the molecular evaluation stage

The two stages are concerned with the generation of feasible molecular structures and the testing of the molecular structures to ensure they meet the requirements issued to the molecule designer. The generate stage involves the selection of functional groups utilized in GCMs, the characterization of groups and the evaluation of molecular feasibility. The test stage involves the selection of group contribution methods for property estimation, calculation of properties, evaluation of property constraints and evaluation of performance indices. The final outcome of the generate and test procedure is a ranked set of product candidates.

2.3.1.2.1. Generate Stage

In this stage, molecules are synthesized by joining functional groups with free-attachments until no free-attachments remain in the generated structure. The free-attachments of a group are the number of chemical bonds available to neighboring groups for attachment (or combination). The free-attachments are also known as their valency. Unlike database search, the generate and test procedure is not confined to molecules already synthesized. But instead, the database may

contain molecules that have not been synthesized yet. Thus, the search is not limited to a given set of molecules. However, if the number of functional groups considered rise then the issue of combinatorial explosion can arise. This is because combinatorially many possible molecular structures will get generated in the database being analyzed.

The functional groups with only one free attachment are defined as terminal groups. All other groups with more than one free attachment are defined as intermediate groups. There are three types of intermediate groups: radial, linear and mixed. The terminal and intermediate groups are combined according to combination and feasibility rules presented in the work of Brignole et al. (1986). The combination rules define allowance of attachments and the feasibility rules ensure that the molecules resulting from the combination of the groups are feasible and can exist in reality. Because of the huge number of combinations, Cismondi and Brignole (2004) introduced an algorithm in order to reduce the size of the problem. Property and structural feasibility constraints have also been proposed in the work of Cismondi and Brignole (2004).

2.3.1.2.2. Test Stage

In this stage, the structures generated in the first stage are evaluated according to the imposed property constraints. The properties that are under bounds are calculated using QSPRs which are property models relating properties to structure. An example of a QSPR is the group contribution model (GCM). In GCMs the property value is calculated by summing the product of the functional group's contribution and number of times the functional group occurs in the molecule of interest.

Besides the aforementioned conventional generate and test procedure, Harper and Gani (2000) introduced a hybrid CAMD method which is based on the generate and test procedure. Their aim

was to rein in the computational cost that is associated with accurate methods. A funnel of sorts is generated in their procedure. The least accurate of the utilized methods to calculate the properties evaluates the maximum number of candidate structures and the most accurate method evaluates the least amount of candidate structures. As the accuracy of methods utilized increases, the error decreases and hence the accuracy of prediction of candidate structures. This reduces the number of candidate structures while simultaneously reducing the overall cost of calculation. Their method consists of three stages:

1. Pre-design – definition phase of the design problem
2. Design – solution phase in terms of generation of feasible molecular candidates
3. Post-design – analysis phase in which the final selection is made

In the pre-design stage, the problem is defined and the aims of the CAMD problem at hand are posed in detail. Then, the properties that need to be evaluated and the evaluation methods for each property are enumerated. Next, the methods for property evaluation as well as the constraints are selected. A problem is then formulated where all available information concerning property and constraint values is taken into account. The main objective of the design stage is to generate feasible candidates that satisfy all the property constraints. These candidates are generated from a set of molecular building blocks (i.e. groups) and then they are tested against the design specifications; the property constraints. This is carried out over four levels, where the input of each level is the output of the previous one. A generate and test approach is applied in each level and the extent of detail varies from the first level, the coarsest level to the fourth level, the most detailed level. At the first level, group vectors are generated from the

combination of groups from a basic set of groups. At the second level, the groups from the first level's vectors are combined to form new feasible molecules, including isomers. An atomistic representation of the new molecules is derived at the third level, where full (atom-based) connectivity information is obtained and the use of property prediction methods based on a higher order of structural descriptors is enabled. Finally, in the fourth level, a 3D representation of the molecules is created, by assigning bond lengths, bond angles and torsion angles. The multi-level structure of the method reduces considerably the number of molecules tested with the most computationally demanding prediction techniques, a fact that makes the method significantly less computationally expensive. In the post-design portion, the molecular candidates created in the design stage are tested against constraints that were not included in previous stages. Those could be more general constraints, such as cost constraints, or environmental constraints. The optimal candidate is selected, based on its performance in all sections of interest. This hybrid methodology has been partially implemented as a computer program, ProCAMD.

2.3.1.3. CAMD Using Mathematical Programming

Unlike the screening and, generate and test approaches detailed earlier, the mathematical programming based approaches for CAMD are an inverse approach. In the screening and, generate and test approaches, the molecular structures part of a database are used to calculate the properties of molecules using property models. Using some pre-defined criteria, the structures are then ranked/evaluated for efficacy in these 'forward approaches'. On the other hand, in the inverse approach of mathematical programming, the property targets are first listed.

These targets can be expressed in terms of property bounds or objectives that need to be maximized/minimized. Using property models, these property targets are translated to targets on the values of occurrence numbers of molecular building blocks. The occurrence number of a building block is the number of times the molecular building block appears in a molecule. A molecular building block can, for example, be a molecular fragment. Signature descriptors, mentioned in section 2.1.3.3., are classified in the category of molecular fragments.

Once the mathematical programming or optimization problem has been set up in terms of the occurrence numbers of molecular building blocks, the problem is solved either using deterministic or stochastic algorithms. In a deterministic algorithm, given a starting candidate solution, the final solution is the same every time the algorithm is implemented. This is because the sequence of steps does not involve any probabilistic variables/operations. On the other hand, in a stochastic algorithm, given a starting candidate solution, the final solution can be different due to involvement of probabilistic variables/operations in the sequence of steps. Obtaining the final solution in both of the algorithms involves the selection of the best values of the occurrence numbers of building blocks that either maximize or minimize the objective property functions of interest to the businesses/end user. The best solution may be an optimal or a near-optimal solution. The best solution can be a near-optimal solution also because property models utilized may perpetuate their errors in the solution space. Thus, due to the errors, the optimal solution may not be the best solution.

Since the occurrence numbers of the molecular building blocks are not continuous, the algorithms used to solve the ensuing optimization problems should have the capability to address

discrete variables. If we are dealing with properties that also depend on continuous variables, for example temperature, the algorithms should additionally be able to deal with continuous variables. It is often the case that nonlinear property models are able to better correlate structures and properties. In such a case, the mathematical programming problem will be formulated as a mixed integer nonlinear programming problem (MINLP). Where 'mixed integer' refers to presence of both discrete and continuous variables in the problem formulation. 'Nonlinear programming problem' refers to the nonlinear nature of the mathematical programming problem due to presence of one or more nonlinear property models. If all the property models utilized are linear then a mixed integer linear programming (MILP) problem will be formulated. It is also possible that some problems consist of multiple objectives. If the objectives are conflicting then a multi-objective optimization problem will be formulated. Akin to MILP and MINLP formulations, multi-objective (MO) formulations, MOMILP and MOMINLP problem formulations will be generated. In MO problems, since conflicting objectives are involved there will not be a single solution that optimizes (maximizes/minimizes) all of the objectives because trade-offs will be involved when an objective(s) is improved. Solutions known as pareto optimal solutions are generated in such a case. Each pareto solution consists of a set of values of objective functions. Each set is an improvement in at least one objective function and worsening of at least one, another objective function.

The mathematical problem that usually arises during CAMD is an MINLP. One of the forms that it assumes is the following:

$$\min/\max F_{obj}(U, V) \tag{2.5}$$

subject to

$$H_1(V) \leq 0 \tag{2.6}$$

$$H_2(V) \leq 0 \tag{2.7}$$

$$H_3(U,V) \leq 0 \tag{2.8}$$

$$H_4(U,V) \leq 0 \tag{2.9}$$

Where, $F_{obj}(U,V)$ is the objective function that either needs to be minimized or maximized. $H_1(V)$, $H_2(V)$, $H_3(U,V)$ and $H_4(U,V)$ correspond to structural feasibility constraints, constraints on pure component properties, property constraints of mixtures and process constraints respectively. Also, U is a vector of continuous variables concerned with processes and/or mixtures. V is a vector of discrete variables concerned with the number of molecular building blocks and/or molecules.

3. Methodology

In this chapter, first, the revised structural constraints are presented that ensure that feasible structures are obtained from the solution of a CAMD problem that utilizes signature descriptors. Next, structural relationships between the reactants and products are developed taking into account whatever the reaction mechanism involved may be. Building upon these, the three design scenarios previously mentioned are addressed. The problem formulation for each of these scenarios is then arrived at. Lastly, tree based ensemble machine learning methods are discussed that can be utilized to develop property models to predict properties of molecules in reactive systems. Specifically, in this work we are interested in predicting the rate constant of a reaction using structures of reactants and solvents. There is a paucity of such models in the scientific literature.

3.1. Revised Structural Feasibility Constraints

In their work, Chemmangattuvalappil et al. (2010), presented a detailed mathematical programming based CAMD framework which utilized signature descriptors. The dominant property to be optimized and the properties that were constrained were expressed in terms of molecular structure using property models. These property models, which were expressed in terms of TIs/GCs, were all treated on a single platform and solved for the occurrence number of atomic signatures. The occurrence number of an atomic signature decides how many times that particular signature appears in the final molecular structure(s). Finally, with these right amounts of atomic signatures, the molecular structures were enumerated by combining these signatures using steps provided in Faulon et al. (2003a). In order for feasible structures to be generated from

the solution, it is necessary to construct structural constraints in terms of occurrence number of atomic signatures. Structural constraints for feasibility have been presented in the work of Churchwell et al. (2004) and Chemmangattuvalappil et al. (2010). However, these constraints are lacking in full consideration of overlap of neighborhoods of bonding atoms. While their CAMD methodologies are applicable when signatures with height greater than 1 are utilized, the overlap of neighborhoods of bonding atoms is not considered beyond the first out-neighborhood in the structural feasibility constraints. Since a signature is a rooted tree that captures the local neighborhood of an atom up to height *h*, it is important to take this fully into consideration while framing structural constraints. Let us consider 2 arbitrary root atoms *X* and *Y*. Let, X_i, atom *X* colored *i* and Y_j, atom *Y* colored *j* have respective signatures of height *h*. Let the respective signatures be $\sigma(X_i)$ and $\sigma(Y_j)$. Now, let there be a bond between atoms X_i and Y_j. Since $\sigma(X_i)$ and $\sigma(Y_j)$ are rooted trees, rooted at X_i and Y_j, X_i and Y_j will also have their respective branches, where the branch at X_i of the tree $\sigma(X_i)$ is the maximal subtree containing X_i as the end node (Balakrishnan and Ranganathan, 2012). The branch at Y_j of the tree $\sigma(Y_j)$ is similarly defined. In this work, the branches at X_i and Y_j will be known as the branches of $\sigma(X_i)$ and $\sigma(Y_j)$. Since X_i and Y_j bond, X_i will appear on the first level of signature of Y_j and vice versa. Let $\sigma^*(Y_j)$ and $\sigma^*(X_i)$ be the height *h* rooted trees of X_i and Y_j respectively, who appear on the first levels of Y_j and X_i respectively. Since X_i appears on the first level of signature of Y_j, the branches of $\sigma(X_i)$ that do not contain Y_j at the child level will overlap with $\sigma^*(Y_j)$ only up to height h-1. However, the branch of $\sigma(X_i)$ that contains Y_j can overlap up to height *h*. This is true vice versa, i.e. when $\sigma(Y_j)$ and $\sigma^*(X_i)$ are made to overlap. Also, it is worth noting that overlapping of the edges (bonds) is also necessary, besides the nodes.

Fig. 3.1 displays height 3 signatures of two atoms C3Bold and N3Bold who form a bond. C3Bold is the C atom shown in bold and has degree 3. Similarly, N3Bold is the N atom shown in bold having degree 3. N3Ital, in italics, is the N3Bold atom that appears on the first level of C3Bold's signature. C3Ital, in italics, is the C3Bold atom that appears on the first level of N3Bold's signature. Which portions of C3Bold's signature of height 3 and C3Ital's rooted tree of height 3, overlap, is shown using numbers adjoining the atoms in signatures. The root atoms have been numbered zero and the atoms in the out-neighborhoods, that overlap, have been numbered progressively from 1 to 3.

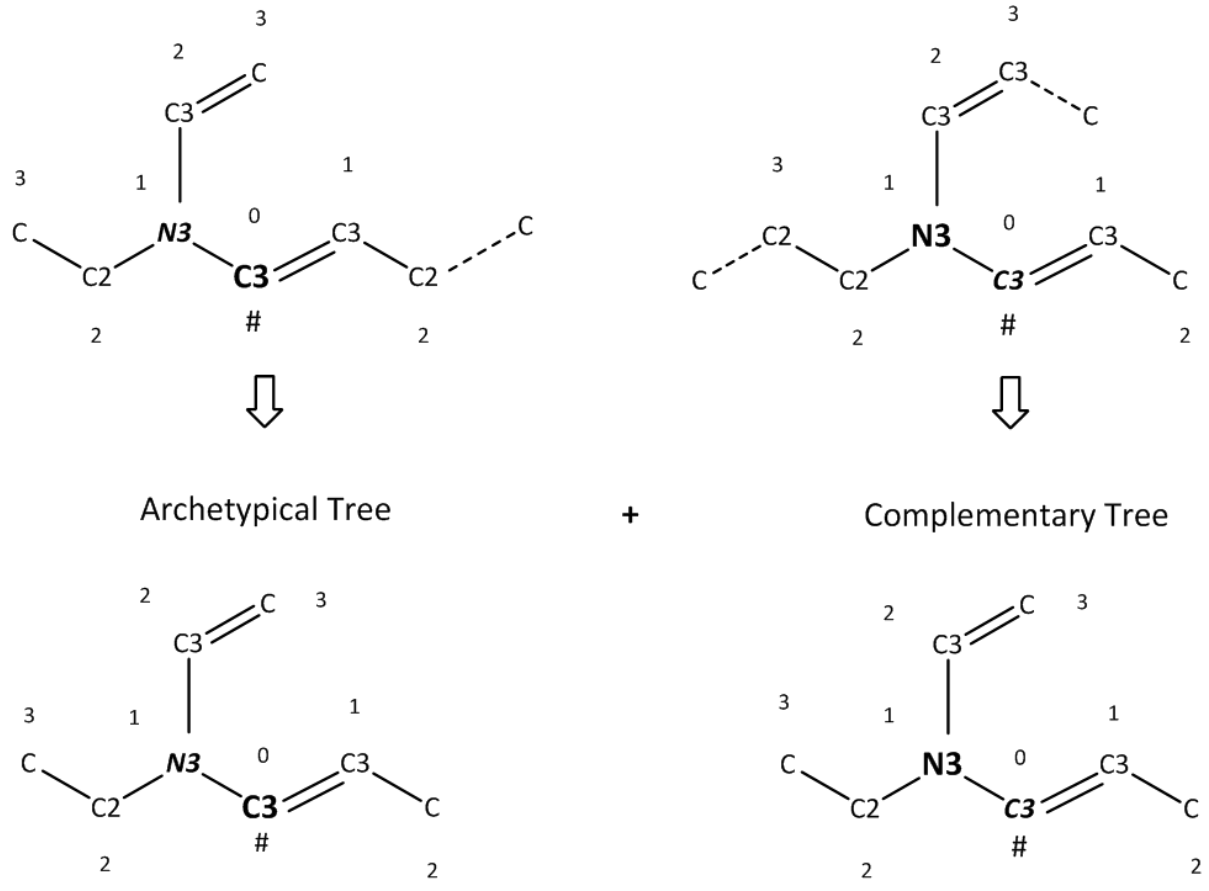

Figure 3.1: Isomorphism of C3's archetypical tree and N3's complementary tree

The terminal atoms with dashed bonds do not overlap. One can observe that the branch of C3Bold's signature not containing N3Ital overlaps with the rooted tree of C3Ital only up to height 2. However, on the branch containing N3Ital, the overlap occurs up to height 3. Vice versa is true also if the N3Bold's signature is made to overlap with the height 3 rooted tree originating from the first level of signature of C3Bold. From here onwards, such an overlap between signatures of bonding atoms will be referred to as 'mutual overlap'. The branches of $\sigma(X_i)$ and $\sigma(Y_j)$ which overlap up to height h-1 only will be referred to as 'non-bonding branches'. The branches of $\sigma(X_i)$ and $\sigma(Y_j)$ having overlap up to height h will be referred to as a 'bonding branch'. $\sigma^*(X_i)$ and $\sigma^*(Y_j)$ will be referred to as 'complementary trees'.

Finally, we can state the sufficient condition for bonding of two atoms as:

X_i and Y_j bond when the rooted tree, obtained by redrawing the non-bonding branches of $\sigma(X_i)$ to height h-1, overlaps with $\sigma^*(Y_j)$. Alternatively, X_i and Y_j bond when the rooted tree, obtained by redrawing the non-bonding branches of $\sigma(Y_j)$ to height h-1, overlaps with $\sigma^*(X_i)$.

$\sigma^*(Y_j)$ and $\sigma^*(X_i)$ thus have height h, but, except one branch all other branches terminate at (h-1)th level. Let the rooted trees obtained after redrawing $\sigma(X_i)$ and $\sigma(Y_j)$ be $\sigma'(X_i)$ and $\sigma'(Y_j)$ respectively. Even these are of height h but except one branch, all other branches terminate at the (h-1)th level. They will henceforth be referred to as 'archetypical trees'. Both the archetypical and complementary trees have been displayed in Fig. 3.1. From the aforementioned discussion, it should be clear that it is important to fully take into consideration the information captured in a root atom's local neighborhood while formulating the structural constraints. The revised

structural feasibility constraints are now presented in the following subsections and compared vis-à-vis those presented in Churchwell et al. (2004) and Chemmangattuvalappil et al. (2010).

3.1.1. Handshaking lemma

In a (molecular) graph, the total number of degrees is twice the number of edges. In order to calculate the number of degrees from an atomic signature, one needs to know only the number of connections a root atom has with the atoms appearing on the first level. Thus, while expressing a handshaking lemma relation, signature heights greater than 1 do not contribute any special information. Thus, the relationship proposed by Chemmangattuvalappil et al. (2010) in terms of occurrence numbers will remain unaltered as it rightly considers the neighborhood up to height one. If the maximum degree is 4 in a molecular graph with *R* no. of circuits, then the handshaking lemma is expressed in terms of the occurrence numbers of signatures as follows:

$$\sum_{i=1}^{4}\sum_{j}^{n_s} D_i x_j = 2\left[\left(\sum_{k}^{N} x_k + \frac{1}{2}\sum_{l=1}^{N_D} x_l + \sum_{m=1}^{N_{2D}} x_m + \sum_{p=1}^{N_T} x_p\right) - 1 + R\right] \tag{3.1}$$

Where, D_i = degree of atom with signature having occurrence number x_j, n_s = number of signatures having degree D_i, N is the total number of signatures in the molecule being designed, and N_D, N_{2D} and N_T are the total number of signatures with root atoms having one double bond, 2 double bonds and one triple bond respectively as part of the first level. The relation presented above can accordingly be modified to include atoms with degree higher than 4.

3.1.2. Conservation of Overlapping Trees

The signature of an atom can be viewed as a directed rooted tree (Jayaseelan et al., 2012). Thus, each bond appearing in the signature, between atoms who will be in adjacent levels, can be represented as an arc. The direction of all the arcs will be away from the root atom. Let us again consider arbitrary atoms *X* and *Y* that have at least one bond between them. In the signature of *X*, *Y* will appear in the first level. Thus, there will be an arc from *X* to *Y*. Similarly in the signature of *Y*, there will be an arc from *Y* to *X* which will now be in the first level. Thus, the same bond between *X* and *Y* will appear as arcs in opposing directions. This information was used to formulate the equations, called consistency equations, presented in the work of Churchwell et al. (2004) and Chemmangattuvalappil et al. (2010). Mathematically this can be stated as:

$$\sum_{p} \eta_p \alpha_p \left[\sigma_p\left(X_{i,root} \rightarrow Y_{j,child}\right)\right] = \sum_{q} \eta_q \alpha_q \left[\sigma_q\left(Y_{j,root} \rightarrow X_{i,child}\right)\right] \quad (3.2)$$

Where, X_i and Y_j are the atoms *X* and *Y* with colors *i* and *j* respectively. $X_{i,root}$ and $Y_{j,root}$ are atoms X_i and Y_j when they are the root. $X_{i,child}$ and $Y_{j,child}$ are atoms X_i and Y_j as children. $\sigma(X_{i,root} \rightarrow Y_{j,child})$ is a signature in which atom X_i is the root and Y_j is the child. Similarly, $\sigma(Y_{j,root} \rightarrow X_{i,child})$ is a signature in which Y_j is the root and X_i is the child. *p* and *q* are the total number of signatures containing arcs of the type $X_{i,root} \rightarrow Y_{j,child}$ and $Y_{j,root} \rightarrow X_{i,child}$ respectively. η_p and η_q are the number of arcs of the type $X_{i,root} \rightarrow Y_{j,child}$ and $Y_{j,root} \rightarrow X_{i,child}$ respectively in the signatures having occurrence numbers α_p and α_q respectively. The signatures have been specified in the square

brackets. However, as mentioned earlier, *X* and *Y* bond only when there is mutual overlap between their respective signatures. Thus, although Eq. (3.2) holds, it can lead to mismatch of signatures that have the potential to bond as it does not capture the information of the local neighborhood beyond the first height. An equation, that addresses this shortcoming, can be expressed mathematically as:

$$\sum_{p} \mu_p^+ \alpha_p \left[\sigma_p\left(X_{i,root} \rightarrow Y_{j,child}\right)^{\#}\right] = \sum_{q} \mu_q^- \alpha_q \left[\sigma_q\left(Y_{j,root} \rightarrow X_{i,child}\right)^{\#}\right] \quad (3.3)$$

Where, $\sigma_p(X_{i,root} \rightarrow Y_{j,child})^{\#}$ is a signature of X_i which has a mutual overlap with signature $\sigma_q(Y_{j,root} \rightarrow X_{i,child})^{\#}$ of Y_j. μ^+_p and μ^-_q are the number of archetypical trees and complementary trees that can be generated from signatures $\sigma_p(X_{i,root} \rightarrow Y_{j,child})^{\#}$ and $\sigma_q(Y_{j,root} \rightarrow X_{i,child})^{\#}$ respectively having occurrence numbers α_p and α_q respectively. It is worth noting that Eq. 3.3 is also valid for the case, $X_i = Y_j = Z_k$, where *Z* is an arbitrary atom colored *k*. This is another difference from the equations presented in Churchwell et al. (2004) and Chemmangattuvalapil et al. (2010). However, there is an exception to this case. If a signature of Z_k, $\sigma(Z_k)$ has a mutual overlap with itself then a bond has a possibility of being established between two Z_k atoms who have same signatures. "Possibility of being established" is being stated because there can be another signature of Z_k that not only mutually overlaps with itself but also with $\sigma(Z_k)$. The existence of other such signatures of Z_k is possible because the non-bonding branches overlap only up to height *h*-1 and

at height *h* in such branches, different atoms can possibly exist. For this separate case, the mathematical expression is:

$$\sum_{r} \mu_r \alpha_r \left[\sigma_r \left(Z_{k,root} \rightarrow Z_{k,child} \right)^{\#} \right] = 2\psi \tag{3.4}$$

Where, Ψ is a non-negative integer. μ_r is the number of archetypical trees that can be generated from the signature $\sigma_r(Z_{k,root} \rightarrow Z_{k,child})^{\#}$ having occurrence number α_r. Both the root and the child levels have atom Z_k.

Although Eq. (3.3) and Eq. (3.4) ensure that signatures present on LHS and RHS mutually overlap, they are concerned only with balancing the total number of archetypical and complementary trees. One needs to additionally ensure, that for all possible signatures of one bonding atom having multiple archetypical trees, only one archetypical tree overlaps with one complementary tree generated from the signature of the other bonding atom. This is the case because when a bond is formed, only one archetypical tree from the possible archetypical trees of a signature overlaps with one of the possible complementary trees generated from the other signature. This can be expressed mathematically as:

$$\sum_{m} \lambda_m^{+} \alpha_m \left[\sigma_m \left(X_{i,root} \rightarrow Y_{j,child} \right)^{\#} \right] \leq \sum_{q} \alpha_q \left[\sigma_q \left(Y_{j,root} \rightarrow X_{i,child} \right)^{\#} \right] \quad (3.5)$$

$$\sum_{n} \lambda_n^{-} \alpha_n \left[\sigma_n \left(Y_{j,root} \rightarrow X_{i,child} \right)^{\#} \right] \leq \sum_{p} \alpha_p \left[\sigma_p \left(X_{i,root} \rightarrow Y_{j,child} \right)^{\#} \right] \quad (3.6)$$

Where, λ^{+}_{m} is the number of archetypical trees of signature $\sigma_m(X_{i,root} \rightarrow Y_{j,child})^{\#}$ and λ^{-}_{n} is the number of complementary trees of signature $\sigma_n(Y_{j,root} \rightarrow X_{i,child})^{\#}$. Signatures $\sigma_m(X_{i,root} \rightarrow Y_{j,child})^{\#}$ and $\sigma_n(Y_{j,root} \rightarrow X_{i,child})^{\#}$ have multiple archetypical and complementary trees respectively. $X_{i,root}$ has *m* signatures with multiple archetypical trees. $Y_{j,child}$ has *n* signatures with multiple complementary trees. Eq. (3.5) and Eq. (3.6) ensure that there exist total number of occurrences of signatures, at least equal to the total number of archetypical and complementary trees belonging to signatures having multiple archetypical and complementary trees. The presented inequalities are the counterpart of the 'parent-child colour inequality' presented in Chemmangattuvalappil et al. (2010). Using Eqs. (3.1), (3.3), (3.4), (3.5) & (3.6) one can thus ensure that infeasible structures are eliminated. Although infeasible structures can be eliminated from the solution pool of a CAMD problem when the previously reported constraints are used, the revised structural constraints will be helpful in the avoidance of generation of such structures in the first place and perhaps improve the search efficiency. Constraints are now developed to address the structural relationship between reactants and products, irrespective of the reaction mechanism and the number of reactants and products involved.

3.2. Molecular Design of Reactants and Products

During molecular design of reactants and products, their molecular structures are varied until the dominant properties of interest are optimized without violation of property and structural constraints. The reaction(s) occurring in the system and their mechanism(s) are known beforehand. Every allowed variation of molecular structures is incorporated in the set of potential atomic signatures. Atomic signatures are generated for each atom present in each reactant and product after consulting their respective general structural formula. In the set of potential atomic signatures, different signatures for the same root atom can exist. The appearance of atoms from variable groups, in the neighborhood of the root atom causes different signatures for the same root atom to exist. A subset is chosen from the set of potential signatures every time a molecular structure is varied. The elements of this subset when combined suitably in correct numbers generate a feasible structure. These numbers are the occurrence numbers of the signatures part of the subset. Depending on the TIs and GCs utilized, the height of the signatures is decided (Chemmangattuvalappil et al., 2010). Usually all signatures are converted to the same height. The height chosen is the highest out of those initially required to capture the structural information encoded in different TIs and GCs appearing in the property models (Chemmangattuvalappil et al., 2010). However, as shown in section 3.3, for a particular case, it might be beneficial to utilize signatures of different heights.

3.2.1. Root Atom Balance of Signatures

In a chemical reaction, atoms of reactants rearrange to yield products. Thus, the number of atoms in a reaction remain constant and the molecular structures of reactants and products are related.

The new position of a reactant atom in the molecular graph of the corresponding product is revealed by the reaction mechanism. Consequently, given a signature from a reactant's set of potential signatures, not only the root atom but atoms in each level of the signature tree can be tracked to the signatures in the products. Thus, the change, if any, in the signatures can be inferred.

In compliance with the involved reaction mechanism, some or all of the atoms in the n^{th} out-neighborhood ($1 \leq n \leq h$) of a reactant atom may appear in the n^{th} out-neighborhood of the repositioned reactant atom. The common atoms may also appear in a different n^{th} out-neighborhood after the reaction. The uncommon atoms can also be tracked from different positions in the reactants. Thus, identification of a product signature obtained from transformation of a reactant signature, with the same root atom, can be carried out. The mentioned root atoms are tracked from molecular subgraphs in reactants to molecular subgraphs in the products to derive relationships between the occurrence numbers of the signatures involved. Let us consider the case where an atom's signature has no variation in the neighborhood after the reaction. If more atoms exist of such kind in a molecular subgraph whose signatures are isomorphic to each other, then they can be summed up together as the occurrence number of that signature particular to that subgraph. It should be mentioned that such signatures can belong to subgraphs part of variable groups and still not get varied after the reaction. In this case, their occurrence numbers will be variable. Since the signatures of such atoms do not vary, their number must remain constant after the reaction as the number of atoms remain constant. One can track the unchanged signatures to molecular subgraphs in products and equate the occurrence values in the reactants to those in these new subgraphs. One can similarly track

isomorphic signatures from reactants to isomorphic signatures of products when there is a variation in the neighborhoods, such that the uncommon atoms in the neighborhood do not originate from variable groups. Since the uncommon atoms in the neighborhood do not originate from variable groups, for every signature in reactant, there is only one corresponding signature in the product. Again, isomorphic signatures of atoms in the reactants belonging to certain subgraphs transforming into set of isomorphic signatures have same occurrence numbers after the reaction.

On the other hand, due to variability of molecular structures, some atoms in reactants will exist whose *n* out-neighborhoods can be filled by different sets of possible atoms. If these different possible sets do not appear in the *n* out-neighborhoods of the repositioned reactant atom then different possible reactant signatures will exist for a corresponding product signature. Conversely, it may happen that different sets of possible atoms can be filled in the *n* out-neighborhoods of the repositioned reactant atom. If they do not originate from the *n* out-neighborhoods of the original reactant atom then different possible product signatures will exist. Thus, to identify the corresponding product signatures obtained after transformation of certain reactant signatures, the atoms common to the *n* out-neighborhoods of the root atom should appear in both reactant and product signatures as mandated by the reaction mechanism. The uncommon atoms if variable will generate different possible reactant and/or product signatures for the same set of common atoms. In this case, even if we have isomorphic sets of possible signatures of root atoms in a subgraph in reactants and products, it will be difficult to attribute the final neighborhoods to the respective atoms. This is because some atoms can have a neighborhood which is distinct from that of other atoms, but is one of those possibilities. Thus, a

balance equation for the occurrences of signatures will be written for each atom separately. In this case the sum of occurrence numbers of possible signatures before and after the reaction will be equal. Also, the sum of occurrences will be equal to 1 since we're writing the balance equation for one atom only. All the aforementioned different cases of alteration (including the lack of alteration) of neighborhoods of atoms can be represented by the following generalized balance equation involving the occurrence numbers, titled as the Root Atom Balance of Signatures (RABS):

$$\sum_{a=1}^{p}\left(n_a \sum_{b=1}^{q} \alpha_{a,b,root\ v}\right) = \sum_{c=1}^{r}\left(n_c \sum_{d=1}^{s} \beta_{c,d,root\ v}\right) \tag{3.7}$$

Where *p* is the number of reactants, n_a is the stoichiometric coefficient of each reactant, and *q* is the number of possible signatures belonging to a reactant that due to the reaction transforms into specific signatures in the products. Also, *r* is the number of products, n_c is the stoichiometric coefficient of each product and *s* is the number of possible signatures in a product that are obtained by transformation of signatures in the reactants under consideration. $\alpha_{a,b,root\ v}$ is the occurrence number of possible atomic signatures of *root v* belonging to one of the reactants. These signatures can transform into possible atomic signatures of *root v* belonging to one of the products having an occurrence number $\beta_{c,d,root\ v}$. *root v* is an atom of the same element and can have different colors associated with the vertex degree in the reactants and products.

Consider the transesterification reaction, shown in Fig. 3.2, as an example to demonstrate how Eq. (3.7) is applied. The numbers adjoining the atomic symbols are the vertex degrees, used as colors. The dashed bond between C4 and O2 is a bond that will break. R, R′ and R″ are assumed to be acyclic saturated groups.

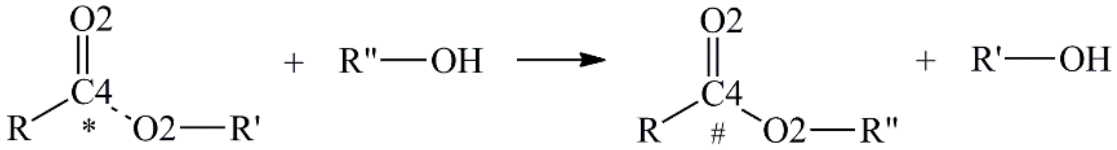

Figure 3.2: Transesterification Reaction

From the mechanism, it is known that the C4 atom (marked *) double bonded to the O2 atom in the reactant ester appears only in the product ester also as C4 (marked #) double bonded to same O2. Let all the possible signatures of height 3 of the reactants and products be generated. Then the root atom balance that can be written for the C4 atom when R is possibly CH_3 i.e. C1 is the following:

α[C4(=O2C1O2(C1))] + α[C4(=O2C1O2(C2(C)))] + α[C4(=O2C1O2(C3(CC)))] + α[C4(=O2C1O2(C4(CCC)))] = β[C4(=O2C1O2(C1))] + β[C4(=O2C1O2(C2(C)))] + β[C4(=O2C1O2(C3(CC)))] + β[C4(=O2C1O2(C4(CCC)))]

Where, *α* and *β* are occurrence numbers of the signatures, in square brackets, of reactant and product C4, respectively. The above holds because according to the mechanism, along with C4, the R group is also transferred to the product ester. Thus, the atoms of R group are common and

will appear as it is in the *n* neighborhoods of C4 in both reactant and product signatures. In this sub-example R has been taken to be possibly C1. So, C1 makes an appearance in each of the signatures. Also, the OR′ group is transferred to the product alcohol and the OR″ group from the reactant alcohol becomes a part of the product ester. These are the variable groups that do not share any common atom. Thus, the branch of the C4 signature tree containing atoms of OR′ can be varied. The same is done for C4 bonded to OR″. Hence, the summation on both sides includes this variation where single bonded O2 is bonded to differently colored atoms. For this example, different variations in R will have different relationships involving signatures rooted at C4. Since in the above reaction only one C4 double bonded to O2 will appear in each ester, an additional condition where the sum of occurrences of all signatures with C4 root atom equals 1 will be imposed on each ester.

Consider atoms in the R group whose neighborhoods do not vary after the reaction. Atoms that do not experience bond breaking or new bond formation will not experience a change in their signatures. Atoms far away from the C4-O2 bond breaking will have unchanged signatures. An example of such a signature is C4(C1C1C1C2(C1C3(CC))). One can notice that the C4-O2 bond is not present in the C4 root atom's neighborhood. Thus, one can say that α[C4(C1C1C1C2(C1C3(CC)))] = β[C4(C1C1C1C2(C1C3(CC)))]. Here, α and β are the occurrence numbers of the signature in the molecular subgraphs R in the reactant ester and product ester respectively. The root atom balance of signatures thus helps in relating the structures of reactants and products in terms of occurrence numbers. This ensures that only those combinations of reactants and products, which are sanctioned by the reaction mechanism, are included in the design process.

3.2.2. Optimizing Dominant Property of Each Product

Now that a methodology has been developed to structurally relate the reactants and products, it will be easier to handle the three design scenarios being addressed in this work. In the first design scenario, we want to generate the structures of reactants and products such that only the optimization of the respective dominant property of each of the products is carried out. Also, each of the products are subjected to their respective set of property constraints. The general methodology for addressing this scenario has been displayed in Fig. 3.3. In this scenario, since the properties of reactants are not constrained, there is no impact on the products through the structural relationships, derived earlier in the form of RABS. Thus, the products can be separately optimized by setting up mathematical programming problems (MPP) for each of the products. Each MPP will consist of the property being optimized, the property constraints and the revised structural constraints derived in section 3.2.1. Since the TIs/GCs can be expressed in terms of occurrence numbers of signatures, the property models utilizing TIs/GCs can be expressed in terms of the occurrence numbers also. The revised structural constraints have also been derived in terms of the occurrence numbers of signatures. Thus, the entire MPP can be formulated in terms of occurrence numbers of signatures. As the occurrence numbers of signatures are non-negative integers, the MPP will be either an MILP or an MINLP problem depending on whether the property models are linear or nonlinear. In this work, the DICOPT solver available in the GAMS software has been used to solve any ensuing MINLP problems. DICOPT tends to be very fast and provides a globally optimal solution for convex functions but for non-convex functions often serves as a successful heuristic approach (Bonami et al., 2012). To solve any ensuing MILP problems, CPLEX has been utilized. The solution obtained by solving the MILP/MINLP problems

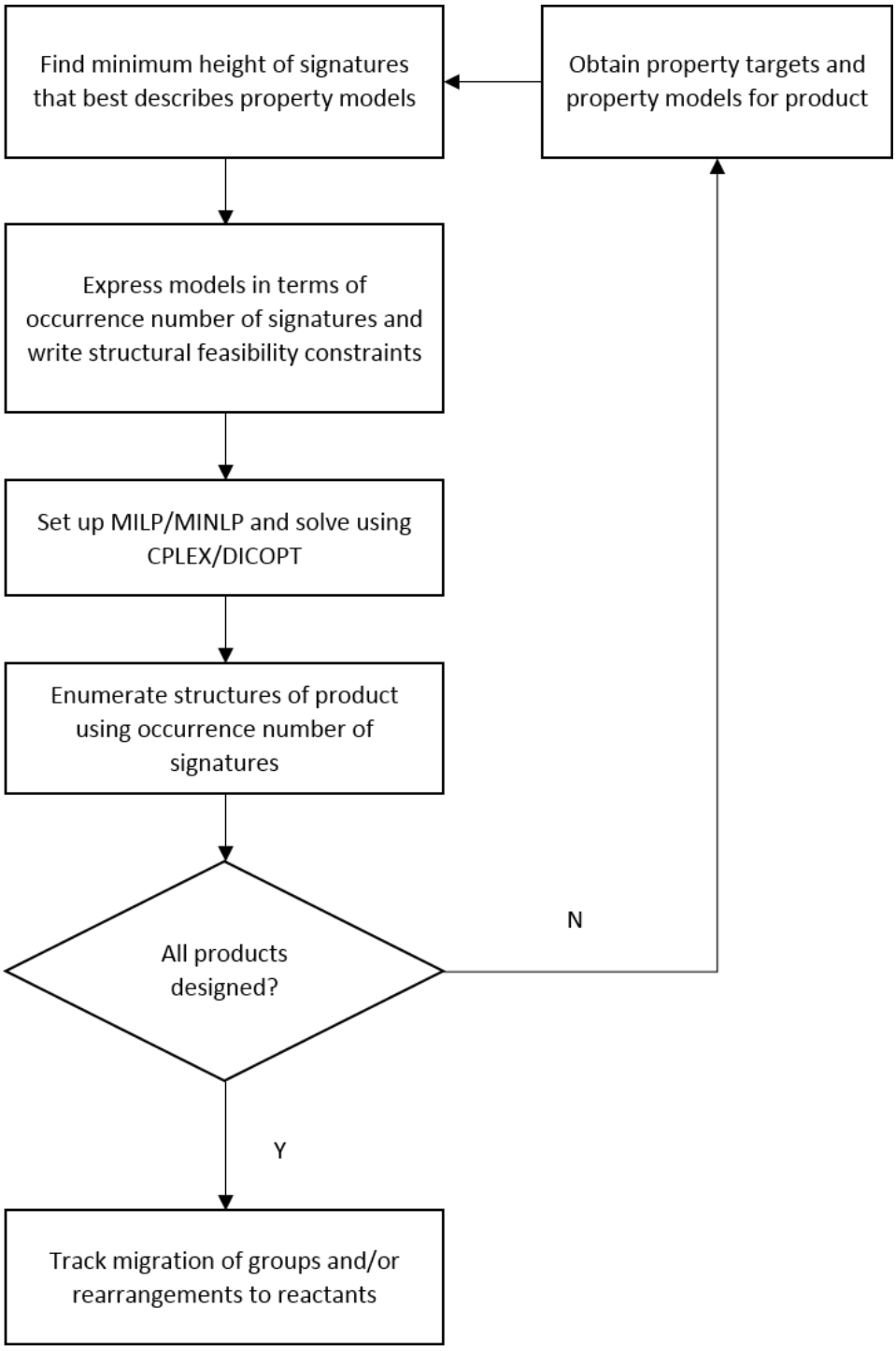

Figure 3.3: General methodology for CAMD of reactants and products in 1st scenario

consists of values of occurrence numbers of signatures of reactants and products. The signatures will be combined in the right numbers using the sufficient condition proposed earlier for bonding of two atoms. This will finally give us the desired structures. Once the structures of the products are determined, using the general structural formula of the reactants and products, and by tracking the groups/atoms, the structures of the reactants can be determined. One can also use the above methodology to design reactants and products when only each reactant's dominant property has to be optimized.

In product design, it is often the case that competing objectives are involved. To accommodate this need, one can also extend the above methodology to include multiple objectives. In this extension, instead of setting up separate MILP/MINLP problems, multi-objective mixed integer linear programming (MOMILP) or multi-objective mixed integer nonlinear programming (MOMINLP) problems will be set up separately for each product. In this work, the augmented ε-constraint method (AUGMECON) has been used to solve the ensuing MOMILP/MOMINLP problems. Details on AUGMECON have been provided in section 3.2.4. Once the set of pareto optimal solutions have been determined, the decision maker (DM) can select the best solution for each product according to his/her requirements. Similar to the case involving single objectives, the groups/subgraphs from final product structures will be tracked to the reactants by utilizing the general structural formulae and the reaction mechanism. In this manner, the final reactant structures will be generated.

3.2.3. Optimizing Properties Dependent on Both Reactant and Product Structures

Since, the methodology developed in section 3.2.2 is applicable only when we want to optimize each product's dominant property, we cannot optimize properties that are dependent on structures of both reactants and products. We will need a different methodology which optimizes such properties, which include thermodynamic properties like Std. Gibbs Free Energy change of a reaction. We would want to optimize properties like Std. Gibbs Free Energy change of reaction to ensure that the reactants and products being designed comprise feasible reactions. Optimization of such properties comprises the second design scenario. A general methodology to address this scenario has been displayed in Fig. 3.4. In this scenario, depending on the linearity or nonlinearity of the properties involved, an MILP or an MINLP problem is formulated to optimize the property of interest. In this CAMD scenario, each of the reactants and products have also been subjected to their respective set of property constraints. Since the properties being optimized depend on structures of both reactants and products, the property constraints of each reactant and product will influence the selection of reactants and products. The MILP/MINLP formulation will thus also include members of all sets of property constraints of reactants and products. Unlike the first design scenario, we are simultaneously considering the variation of the reactant and product structures due to their combined influence on the property being optimized. Since structures of reactants and products are related and since we are considering their variation simultaneously, the RABS relationships will also be employed in the MILP/MINLP formulation. This will ensure their selection in accordance with the reaction mechanism. In addition, the revised structural constraints will be included for each reactant and product to ensure their structural feasibility.

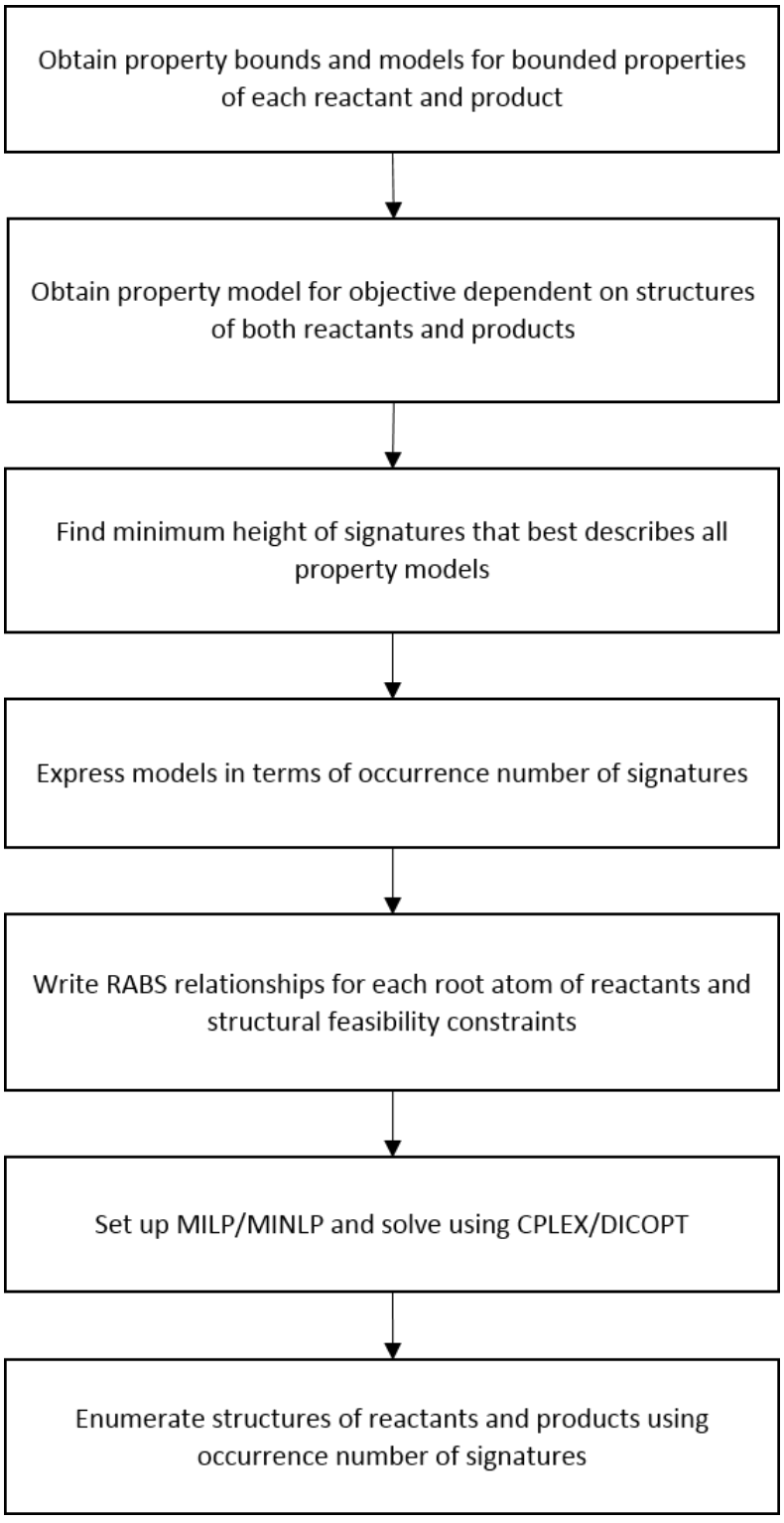

Figure 3.4: General methodology for CAMD of reactants and products in 2nd scenario

Like the first scenario, we have utilized the DICOPT solver to solve any ensuing MINLP problem and CPLEX for any ensuing MILP problem. The solution obtained by solving the MILP/MINLP problem consists of values of occurrence numbers of signatures of reactants and products. The signatures will be combined in the right numbers using the sufficient condition proposed earlier for bonding of two atoms. This will finally give us the desired structures.

Unlike the first design scenario, we are not formulating separate MILP/MINLP problems but a single MILP/MINLP problem. However, if we want to incorporate additional objective functions of any of the reactants and/or products, some of which are conflicting, then a single MOMILP/MOMINLP problem will be formulated instead. This is again distinct from the multi-objective case considered in section 3.2.2. The addition of conflicting objectives of any of the reactants and products is possible because already the MPP consists of a property which is dependent on the structures of each reactant and product. In our work, we have used AUGMECON to solve the ensuing MOMILP/MOMINLP problems.

3.2.4. Optimizing Dominant Properties of Each Reactant and Product

In the third design scenario, each reactant and product has dominant properties to be optimized. Also, each reactant and product has been subjected to a set of property constraints. In the second scenario we have already allowed for the possibility of addition of dominant properties of any of the reactants and products. However, what will be the outcome if we remove the property which is dependent on the structures of each reactant and product? Can the third design scenario be still formulated as a multi-objective optimization problem? As one will find out, from the

following discussion, the problem can still be formulated as a multi-objective optimization problem when conflicting objectives are involved of each reactant and product.

To develop a methodology to address this third CAMD scenario, let's consider a general reaction of the following type:

AB + CD → AC + BD

Where, A, B, C and D are the molecular subgraphs of the structurally variable reactants AB and CD that get exchanged due to the reaction to yield AC and BD. Using RABS, one can relate the occurrence number of signatures of atoms that are part of A of AB to the occurrence numbers of signatures of atoms of A of AC. Similarly, AB can be structurally related to BD, CD to BD and CD to AC and vice versa. Now, in our work we are concerned with properties that are related to structure. These properties are expressed in terms of occurrence numbers of atomic signatures which are non-negative integers. So, if some molecules are structurally related, a variation in the properties of one molecule will vary the properties of the others as well. In our case if we place property constraints on AB, constraints on the occurrence number of signatures of atoms of A and B will be placed. Now, AC has property constraints of its own which places restrictions on signature occurrence numbers of both A and C. Also, the allowed values of signature occurrence numbers in A influence the allowed values of signature occurrence numbers in C. Since A appears in both AB and AC, constraints on A due to constraints on AB will influence both the selection of A and C. Since structure translates to properties through property models, properties that are constrained are also influenced. An inverse effect of influence of property constraints of AC on occurrence number of signatures of A in AB will also take place by a similar argument. Similarly

constraints on AB and CD will influence BD's constraints and vice versa, and CD's constraints will influence those of AC and vice versa. Thus, property constraints of each of the reactants and products influence each other. Same set of arguments can be used to show the influence of property objective functions of each of the reactants and products on themselves. Thus, when the property objective functions of some of the reactants and/or products are conflicting, a multi-objective optimization problem will have to be set up. The analysis above can be carried out for a different type of general reaction as well and similar conclusions can be accordingly derived. The general methodology to address this third CAMD scenario has been shown in Fig. 3.5.

Let us consider an instance of the aforementioned reaction. In this example, D is the Hydorgen atom, H.

$AB + CH \rightarrow AC + BH$

Also, AB, CH, AC and BH's property objective functions, P(AB), P(CH), P(AC) and P(BH) are to be maximized. AB, CH, AC and BH's respective set of properties are also constrained. Let the output of P(AB), P(CH) decrease with the increase in the value of the inputs, i.e. the occurrence numbers of signatures. Let the output of P(AC) and P(BH) increase with the value of the inputs. Let us increase the value of occurrence number of signatures of atoms in A, B and C. This will cause P(AC) and P(BH) to rise. P(AB) and P(CH) however will reduce. Thus, there exists a conflict between the objectives. The variation in the occurrence numbers of signatures of atoms in A of AB is subject to the property constraints of AB. Since A appears in AC also, the property constraints on AC impact the occurrence numbers of signatures of atoms in A also. Since property constraints have been placed on AB and not A and B individually, occurrence numbers of

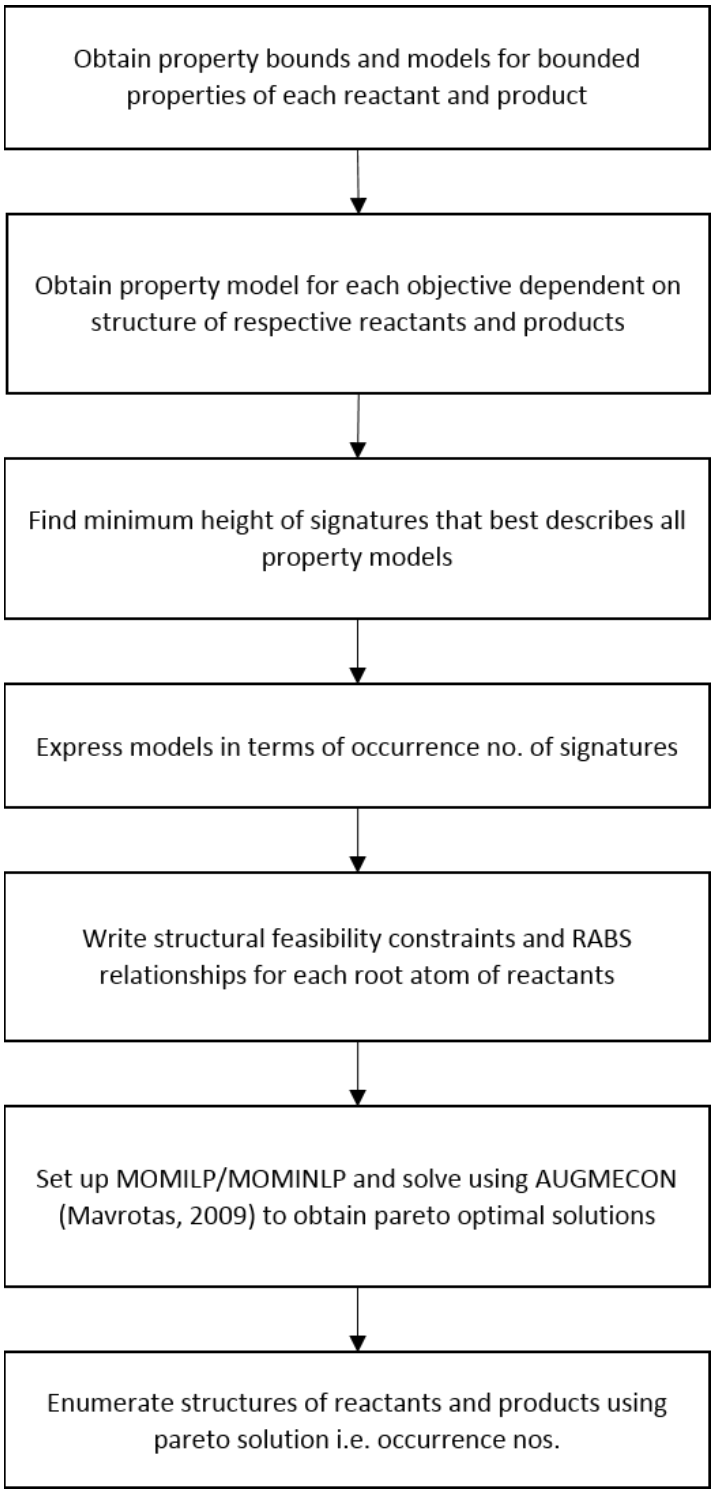

Figure 3.5: General methodology for CAMD of reactants and products in 3rd scenario

signatures of atoms in B depend on the values of occurrence numbers of signatures of atoms in A. Thus, the rise in occurrence numbers of signatures of atoms in B will occur such that the property bounds on AB are not violated. Similarly, property constraints on BH will also affect the occurrence numbers of signatures of atoms in A. This is because B of AB appears in BH and the values of occurrence numbers of signatures of atoms in B is impacted by property constraints on BH. This will further restrict the rise in values of occurrence numbers of signatures of atoms in A since the property constraints on AB cannot be violated. After using similar arguments, one can deduce that the rise in occurrence numbers of signatures of atoms in AB, CH, AC and BH are impacted by all the property constraints. The impact of objective functions can similarly be argued. Thus, a single multi-objective problem will be formulated that includes all the objectives and property constraints. Although we have assumed a rise in the values of occurrence numbers of signatures of atoms in A, B and C, a reduction in some or all of the occurrence numbers also yields conflicting objectives. The analysis conducted here can also be used to check if the objectives are conflicting when some or all of the property objective functions are minimized. The analysis will also have to take into account if the nature of property objective functions is different from what has been assumed here. It is worth noting the distinction of the third design scenario from the first. In the first design scenario, separate optimization problems were set up for each of the products. Here, only a single optimization problem is being set up. Removal of the property constraints and the objective functions of the reactants from the design problem will yield the first design scenario. In the third design scenario, it can be noticed that the influence of structure of reactants on properties of products and vice versa exists irrespective of the linearity or nonlinearity of the property models utilized. Thus, if nonlinear models are involved, the design

problem is formulated as a multi-objective mixed integer nonlinear programming problem (MOMINLP). In this formulation, the objective functions include all the conflicting property objective functions of each of the reactants and products. The MOMINLP problem also includes, like the second design scenario, property and revised structural constraints of each reactant and product. Additionally, the RABS relationships for each root atom are included to ensure that only those combinations of reactants and products are evaluated who comply with the reaction mechanism.

To solve multi-objective mathematical programming problems, the available methods can be categorized as no-preference methods, *a priori* methods, interactive methods or *a posteriori* methods (Miettinen, 1998). However, these distinctions are not rigid. Each of these methods have their advantages and drawbacks. Depending on the nature of the problem and the stage of decision making, with regards to preferences of objectives, in the solution process, by the decision maker (DM), one can appropriately choose a method. The advantages and drawbacks of these methods have been discussed in Miettinen (2008) and Miettinen et al. (2008). In this work, we will be relying on the augmented ε-constraint method (AUGMECON) (Mavrotas, 2009). It is an *a posteriori* method. In such a method, the DM is involved in the solution process after the pareto optimal values have been generated. Thus, such methods offer the advantage of providing a broad picture of the trade-offs involved before a choice is made by the DM. By increasing the number of representative pareto optimal solutions generated, the DM's confidence can be increased. With such methods, generally, a concern can be the computational time involved and the lack of widely available software (Mavrotas, 2009). AUGMECON however alleviates these issues. In an ε-constraint like optimization scheme relying on early exit from loops, for increased

efficiency, the pareto optimal set is populated using AUGMECON. AUGMECON guarantees generation of pareto optimal solutions which is not the case with the conventional ε-constraint method. In case of MOMINLP problems, the guaranteed pareto optimal solutions may be locally pareto optimal. But, this is better than the possibility of generating local weak pareto optima using the conventional ε-constraint method. We once again rely on the DICOPT solver in GAMS to solve the MINLP problems which are part of the AUGMECON scheme. AUGMECON in addition to supported pareto optima, can generate unsupported pareto optima also. Supported optima lie on the boundary of the feasible objective space while unsupported optima do not. The number of representative optima obtained from AUGMECON can be controlled by adjusting the number of grid points in each of the objective function ranges. If the problem posed in the third design scenario can be formulated as a multi-objective integer programming (MOIP) problem then the improved version of AUGMECON, AUGMECON2 (Mavrotas and Florios, 2013) can be utilized. In the MOIP problem, only linear constraints and linear objective functions exist. AUGMECON2 has been demonstrated to be much more efficient than AUGMECON in solving such kind of problems (Mavrotas and Florios, 2013; Florios and Mavrotas, 2014).

3.3. Reduction in Number of Signatures Generated

As discussed before, all the property models, the structural feasibility constraints and the RABS relationships are expressed in terms of occurrence numbers of signatures. As mentioned in section 3.2.1, all the occurrence numbers of smaller height signatures are expressed in terms of occurrence numbers of signatures of biggest height. This can be mathematically expressed as:

$$\alpha[{}^{h}\sigma] = \sum_{i} \alpha[{}^{H}\sigma_i] \tag{3.8}$$

Here, ${}^{h}\sigma$ is the signature of lower height, h. ${}^{H}\sigma_i$ is a signature of higher height, H, which when redrawn to height h is isomorphic to ${}^{h}\sigma$. α is the occurrence number of the signature mentioned in the square brackets. Eq. (3.8) is valid because the signatures of higher height can be generated from signatures of lower height, having same root atom, by adding different possible combination of atoms in the k^{th} out-neighborhoods. Here, $h < k \leq H$.

With increasing height, a concern that can arise is the increment in the number of possible signatures that need to be generated. As the number of out-neighborhoods increases, more combinations of atoms can be introduced in the signature. Hence, the increment in the numbers. With the rise in number of signatures, the number of structural constraints required to solve the ensuing MINLP problems also rises. Usually when GCMs are utilized, one is not required to generate all possible signatures to calculate the property value of a molecule. Only those signatures need to be generated which can effectively capture the structure of the utilized functional groups in the neighborhood of one of the atoms in the functional group. In a GCM, one can thus replace the occurrence number of a functional group with the occurrence number of an equivalent signature. If the highest height stems from the presence of a group(s) in a GCM instead of a TI in a QSAR/QSPR, then one can reduce the number of highest height signatures that need to be drawn significantly. A step wise strategy that can be used in such a case is the following:

1. Draw all the signatures that represent groups requiring highest height.

2. Draw all possible signatures with the second highest height.

3. The signatures of highest height can be redrawn to give signatures of second highest height. Hence, subtract the occurrence number of the signature of highest height from the occurrence number of equivalent signature of lower height. This is in accordance with Eq. (3.8). This effective occurrence number is the actual occurrence number of such equivalent signatures when they do not represent the particular equivalent signature with the highest height. If there are more signatures, having the same equivalent signature of second highest height, their occurrence numbers will be subtracted too.

4. Express groups and TIs requiring signatures of lesser heights in terms of the occurrence number of signatures of second highest height. The occurrence number will be the sum of effective occurrence number and the occurrence number of equivalent signatures of highest height.

5. Write the structural feasibility constraints using highest height signatures only.

6. Write the structural feasibility constraints using second highest height signatures and highest height signatures. The highest height signatures are also being added because their occurrence number is part of total occurrence number of equivalent signature of second highest height.

7. When a signature of height *h* has a mutual overlap with another signature of height *h*, the non-bonding branches overlap only up to height *h*-1. In the case being considered here, some

of the signatures are of a higher height. When there is a height disparity, the lower height signature can overlap with the height *h* signature originating from the first level of the signature with higher height. Hence, write extra structural constraints of the following form:

$$\sum_{m} \alpha_m [{}^{H'}\sigma_m \left(Y_{j,root} \rightarrow X_{i,child}\right)^{\#}] \leq \alpha_q [{}^{h'}\sigma_q \left(X_{i,root} \rightarrow Y_{j,child}\right)^{\#}] \quad (3.9)$$

Where, ${}^{h'}\sigma(X_{i,root} \rightarrow Y_{j,child})^{\#}$ is the signature of lower height, *h´*, which overlaps with the height *h´* signature originating from the first level of ${}^{H'}\sigma_i(Y_{j,root} \rightarrow X_{i,child})^{\#}$. α is the occurrence number of the signature mentioned in the square brackets. The above inequality needs to be written because the structural feasibility constraints for the second highest height only consider overlap of non-bonding branches up to height (*h´-1)*. Also, Eq. (3.9) is an inequality instead of an equality because ${}^{h'}\sigma(X_{i,root} \rightarrow Y_{j,child})^{\#}$ can mutually overlap with other signatures of height *h´*. Here, it is being insured that when signatures on LHS are selected, there are enough signatures on RHS that can lead to bond formation between their root atoms.

8. In spite of the above structural constraints, the disparity in the height of signatures may lead to generation of infeasible structures due to mismatch. Thus, eliminate infeasible structures from the solution pool. This is simpler than generating all signatures of highest height along with the corresponding structural constraints.

3.4. Tree-Based Ensemble Machine Learning

In the previous sections we have developed methodologies to design reactants and products in different scenarios. However, if we were to use these methodologies to design reactants and products such that the rate of reaction is optimized, one will realize that currently not many reliable models are available that relate the rate constant to structures of reactants and solvents. In order to improve upon the currently available models and also in order to offer scalable methods, we explored the use of ensemble machine learning in our work. Specifically, we were interested in decision tree based methods because ensemble learning works particularly well when they are based on decision trees (Rokach, 2016). It has been shown, repeatedly, that ensemble learning improves the predictive performance of a single model. Tree based ensemble learners are frequently used in many real-world applications in domains such as engineering, information retrieval and medicine. With respect to QSPR development, however, ensemble learning, in general, is still under-represented in the literature (Kew and Mitchell, 2015).

Ensemble machine learning methods, as the name suggests, involve the generation of ensemblea of models, generally trained on sampled datasets. These models are then combined in some form (for example, by averaging) to then provide predictions of response variables of interest. Contrary to expectations, many ensemble learning methods aim to generate constituent models that can predict barely better than what can be achieved by chance. The ensemble methods rely on such models because they are computationally inexpensive to generate. Such models are known as ‘weak learners’ or ‘base learners’. A large number of such inexpensive models can then be combined together using various methodologies to offer reliable predictions. Two of the most common approaches to generating ensemble models are discussed as follows:

3.4.1. Randomization-Based Approaches

Randomization-based methods produce different models from a single initial learning sample by introducing random perturbations into the learning procedure. A commonly utilized randomization-based approach is bagging. Bootstrap aggregation or bagging is a general-purpose procedure for reducing the variance of a machine learning method. Given a set of n independent observations $Z_1, \ldots Z_n$, each with variance σ^2, the variance of the mean $\hat{Z}$, of the observations is given by σ^2/n. Thus, averaging a set of observations reduces variance. Thus, extending this to machine learning models, one can obtain predictions with reduced variance contribution to error, if we aggregate various models and average their predictions (James et al., 2013b). The models can be trained on different samples drawn from the population. However, in a real world scenario, it can be difficult to ascertain the sampling distribution of the population. Often one is limited to small datasets. A strategy to address this problem is the generation of multiple datasets by bootstrap sampling. Bootstrap sampling involves the generation of datasets of the same dimensions as that of the original dataset. However, the sampling is carried out with replacement. Once these samples are generated base learners are trained on these bootstrap samples. The predicted value of the response variable is then obtained by averaging the response values obtained from the base learners. This process is visualized in Fig. 3.6 (Liang et al., 2011).

3.4.1.1. Random Forests

Random forests are an innovation based on bagged decision trees which are shown in Fig. 3.7. Bagged trees are generated by aggregating trees trained on bootstrap samples. Random forests are different from bagged trees in that they allow for split-variable randomization i.e. at each

split of the decision tree, only *m* out of *p* total predictors are searched (Breiman, 2001). Building a random forest involves the following:

Inputs:

- input data (x_i, y_i), where $i = 1, \ldots, p$
- number of iterations *B*, here, number of bootstrap samples generated
- choice of the loss function $\varphi(y, f)$, where *f* is the true function relating *y* and *x*

Algorithm:

1. for $b = 1$ to *B* do
2. create training set d_b by bootstrap sampling of input data
3. grow regression tree $\hat{r}_b(x)$ using d_b by sampling m of p descriptors at each split

The random forest fit at any prediction point, say x_0, can then be computed as the following average:

$$\hat{r}_{rf}(x_0) = \frac{1}{B}\sum_{b=1}^{B} \hat{r}_b(x_0) \tag{3.10}$$

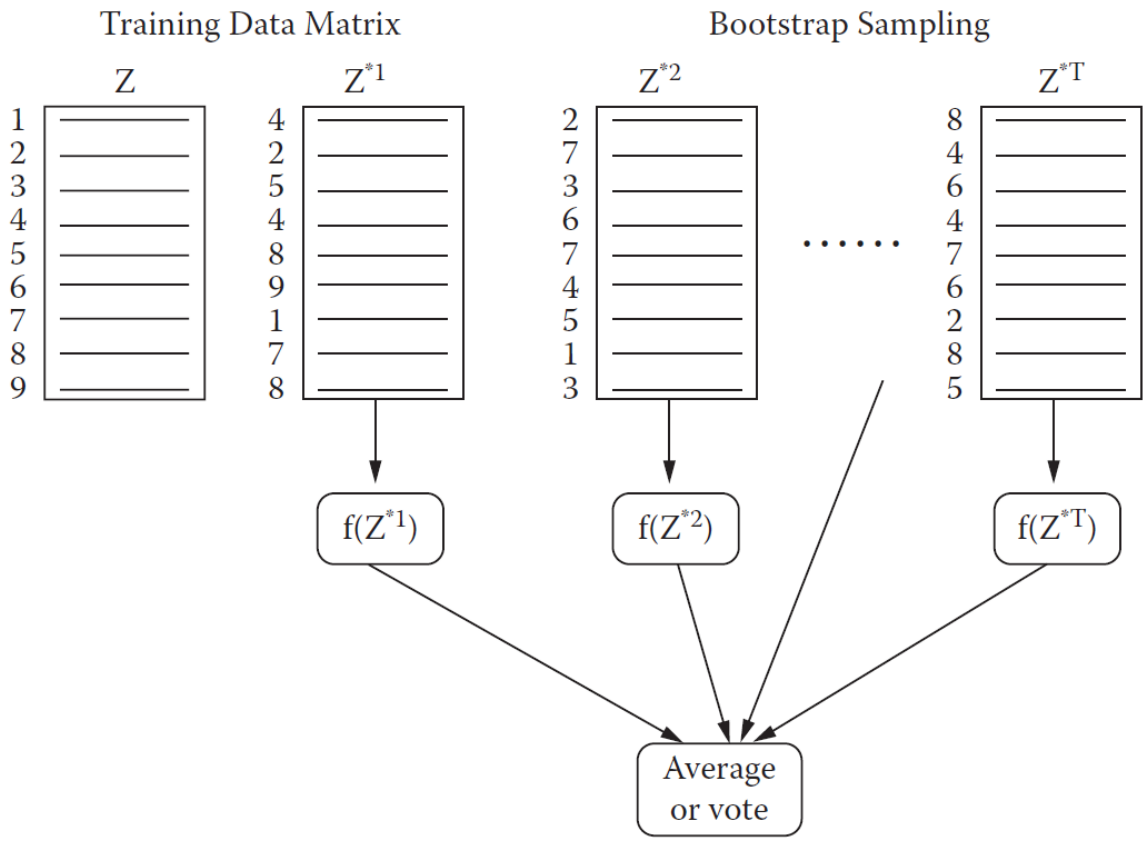

Figure 3.6: Averaging of models trained on bootstrap samples Z^{*1}, Z^{*2}, …, Z^{*T}

3.4.1.2. Regularized Random Forests

Regularized random forests, a recent extension of random forests, apply a regularization framework to random forests and can select a compact feature subset (Deng and Runger, 2013). Regularization usually involves the addition of a penalty to a loss function in order to prevent overfitting. In the case of regularized random forests, a penalty coefficient is utilized while evaluating features that split a node of the decision tree in a random forest model.

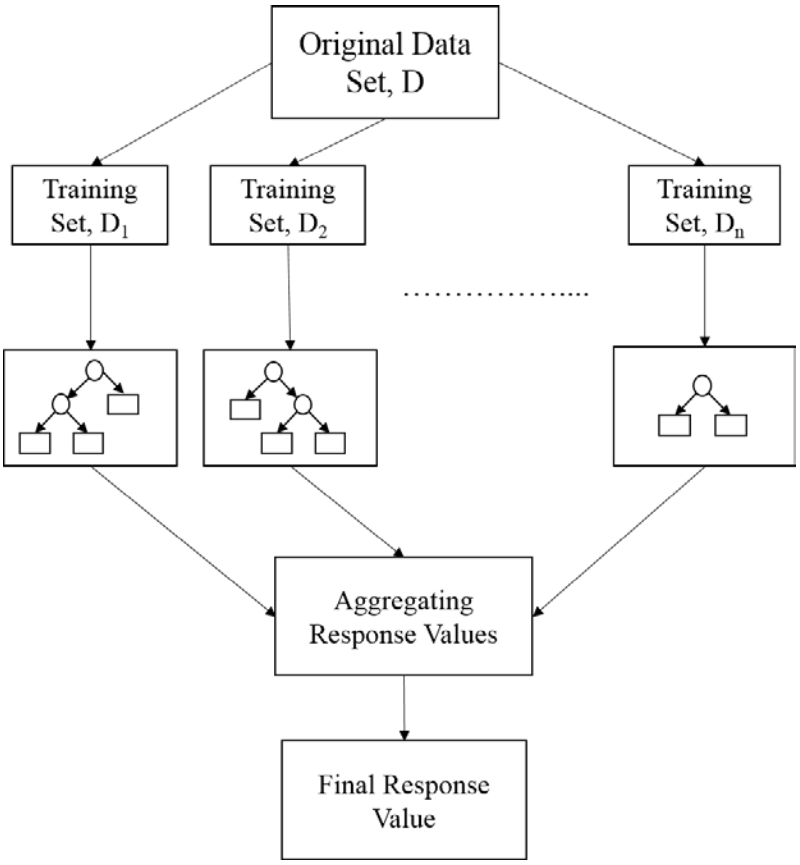

Figure 3.7: Bootstrap aggregation of decision trees

3.4.1.3. Extremely Randomized Trees

Extremely randomized trees (ERTs), a recent innovation in ensemble learning, differ from random forests in that they utilize the whole sample for generating constituent decision trees instead of bootstrap replicas (Geurts et al., 2006). ERTs essentially consist of randomization of both the attribute and the cut-point choice while splitting a tree node. This helps in the reduction of the error associated with variance. Relying on the whole sample helps in the reduction of bias.

3.4.2. Boosting

Boosting works by choosing training sets for base learners in such a fashion so as to force them to infer something new about the data each time they are called (Schapire, 2003; Meir and Rätsch, 2003). While bagging can be thought of as a parallelization of model fitting by generating base learners from sampled sets, boosting is a sequential approach to construction and addition of models to the ensemble (James et al., 2013b). At each iteration, a new weak, base-learner model is trained with respect to the error of the whole ensemble learnt so far. The construction of each base learner is strongly dependent on the previously generated base learners. Like bagging, boosting is a general approach that can be applied to many machine learning problems.

3.4.2.1. Gradient Boosted Regression Trees

Ensemble methods based on gradient-descent based formulation of boosting are termed gradient boosting machines (GBMs) (Natekin and Knoll, 2013). The GBM proposed by Friedman (2001) consists of the following:

Inputs:

- input data (x_i, y_i), where $i = 1, \ldots, m$
- number of iterations N
- choice of the loss function $\varphi(y, f)$, where f is the true function relating y and x
- choice of the base learner model $h(x,\vartheta)$, where ϑ represents the model parameters

Algorithm:

1. initialize $\hat{f}_0$ with a constant
2. for t=1 to N do:
3. compute the negative gradient $g_t(x)$
4. fit a new base-learner function $h(x, \vartheta_t)$
5. find the best gradient descent step size ρ_t
6. update the function estimate:
7. $\hat{f}_t \leftarrow \hat{f}_{t-1} + \rho_t h(x, \vartheta_t)$
8. end for

In the algorithm listed above, $\hat{f}$ represents the model, an estimate of f. Also, ρ_t is calculated using the following formula:

$$\rho_t = \arg min_\rho \frac{1}{B} \sum_{i=1}^{m} \varphi[\, y_i, \hat{f}_{t-1}(x_i) + \rho h(x_i, \theta_t)] \tag{3.11}$$

In our work, h were regression trees. In general, it is worth noting that, with ensembles of decision trees, an issue that can arise is model interpretability. This is because a large number of trees are generated in which the training data is encoded. Conventional models with parameters are not generated by decision tree algorithms. Hence, tree based methods are categorized as nonparametric machine learning methods.

4. Case Studies

To exemplify and demonstrate the effectiveness of the aforementioned methodologies for each design scenario, we solve case studies, one each for each design scenario.

4.1. Case Study 1

We consider the transesterification reaction that is portrayed in Fig. 3.2. The aim of this case study is to design reactant ester (RCOOR′) and reactant alcohol (R″OH) that generate product ester (RCOOR″) and product alcohol (R′OH) with respective optimal flash points (F_p). The flash points are to be maximized given that the boiling point (T_b) and log(P) values of RCOOR″, and the T_b and $log(LC_{50})$ values of R′OH are constrained. Here, P is the octanol-water partition coefficient and LC_{50} is the acute toxicity. The property ranges are listed in Table 4.1. The property models utilized have been listed in Table 4.2. The F_p GCM was developed by Hukkerikar *et al.* (2012). In the GCM, C_i, D_j and E_k are contributions of first, second and third order groups of type i, j and k respectively. N_i, M_j and O_k are their respective occurrence numbers. The T_b GCM (Hukkerikar *et al.*, 2012) has the same form as GCM of F_p. In the log(P) model (Šoškić and Plavšić, 2005), ${}^1\chi^{opt}$ is the optimized first-order molecular connectivity index. I_{MET}, I_{PHYD}, I_{ALRIN} and I_{CONJUG} are indicator variables (IVs) for compounds containing methyl group attached to heteroatoms, H atom bonded to strongly electronegative group, an aliphatic ring and conjugation respectively. I_{HG2} and I_{HG3} are IVs representing 2 and 3 geminal halogens on a C atom, I_{HVIC} represents halogens bonded to vicinal carbons and I_{PG2} and I_{PVIC} represent polar groups separated by 1 and 2 carbons respectively. In the log(LC_{50}) QSAR (Juric *et al.*,1992), ${}^0\chi^v$ is the zero-valence connectivity index. R, R′ and R″ in the esters and alcohols have been assumed to be homogeneous, acyclic and

saturated. The highest degree of atoms in these groups has been restricted to 3. This puts a limit on the structures being explored in the chemical space but does not affect the CAMD methodology. We have also restricted ourselves to first order groups only. There are no third order groups for our case. If we ignore the second order groups then the maximum height of utilized signatures is 2. We expressed all groups and TIs in terms of occurrence number of signatures of height 2 after ignoring the second order groups. Sizeable errors could get introduced after using such an approximation of GCMs. With the aforementioned assumptions taken into account, signatures of the atoms that make the product ester and alcohol have been listed in Table 4.3 and Table 4.4 along with their occurrence number variables. The '(*)' adjacent to certain atoms in signature strings signifies that are no further connections at a higher height. Now, in accordance with section 3.2.2, for each product a separate optimization problem was set up. Beside the property constraints, the revised structural constraints were also included in the problem formulation. For the product alcohol, the ensuing MINLP was solved using DICOPT and for the product ester, the ensuing MILP was solved using CPLEX. Some of the solutions obtained after solving the two optimization problems are shown in Table 4.5. Now, since we know the mechanism of the reaction and the general structures of the products, we can trace the determined R, R′ and R″ in the final product structures to the reactants. Let us for example consider, solution no. 8. For this case, R is $CH_3CH_2CH_2CH_2CH(CH_3)$, R′ is CH_3CH_2 and R″ is $CH_3CH_2CH_2$. Since R of RCOOR″ appears in RCOOR′, R″ of RCOOR″ appears in R″OH and R′ of R′OH appears in RCOOR′, the final structures of the reactants can be determined. These have been displayed in Table 4.5. For our case study, only one optimal structure of the product alcohol

was obtained i.e. ethanol. However, multiple structures were obtained corresponding to the optimal property of the product ester.

Table 4.1: Property Constraints on Products

Products	Properties	Upper Bound	Lower Bound
RCOOR˝	F_p (K)	Maximum	
	T_b (K)	485	395
	log(P)	5	-
R´OH	F_p (K)	Maximum	
	T_b (K)	415	320
	log(LC_{50})	-	-1

Table 4.2: Property Models

Properties	Property Models
F_p (K)	$F_p - F_{p0} = \sum_i N_i C_i + \sum_j M_j D_j + \sum_k E_k O_k$

T_b (K)	$\exp(T_b/T_{b0}) = \sum_i N_i C_i + \sum_j M_j D_j + \sum_k E_k O_k$
log(*LC50*)	$\log(LC_{50}) = 2.975 + 1.169 \times \log({}^0\chi^v) - 7.309 \times (\log({}^0\chi^v))^2$
log(*P*)	$\log(P) = 0.829 + 1.055 \times ({}^1\chi^{opt}) + 0.580 \times I_{MET} + 0.367 \times I_{PHYD} - 0.627 \times I_{ALRIN} + 0.454 \times I_{CONJUG} + 0.658 \times I_{HG2} + 1.726 \times I_{HG3} + 0.381 \times I_{HVIC} + 1.271 \times I_{PG2} + 0.605 \times I_{PVIC}$

Table 4.3: Signatures of Product Ester

ATOM TYPES	OCCURRENCE NOS. (yi)	SIGNATURES
O2 (Single Bonded)	y1	O2(C4(=OC)C1(*))
	y2	O2(C4(=OC)C2(C))
	y3	O2(C4(=OC)C3(CC))
C at Parent Level of O2 (Single Bonded)	y4	C1(O2(C))
	y5	C2(O2(C)C1(*))
	y6	C2(O2(C)C2(C))
	y7	C2(O2(C)C3(CC))
	y8	C3(O2(C)C1(*)C1(*))
	y9	C3(O2(C)C2(C)C1(*))
	y10	C3(O2(C)C2(C)C2(C))
	y11	C3(O2(C)C3(CC)C1(*))
	y12	C3(O2(C)C3(CC)C2(C))
	y13	C3(O2(C)C3(CC)C3(CC))
C at Child Level of O2 (Single Bonded)	y14	C1(C2(O))
	y15	C1(C3(OC))
	y16	C2(C2(O)C1(*))
	y17	C2(C2(O)C2(C))
	y18	C2(C2(O)C3(CC))
	y19	C2(C3(OC)C1(*))
	y20	C2(C3(OC)C2(C))
	y21	C2(C3(OC)C3(CC))
	y22	C3(C2(O)C1(*)C1(*))
	y23	C3(C2(O)C2(C)C1(*))

	y24	C3(C2(O)C2(C)C2(C))
	y25	C3(C2(O)C3(CC)C1(*))
	y26	C3(C2(O)C3(CC)C2(C))
	y27	C3(C2(O)C3(CC)C3(CC))
	y28	C3(C3(OC)C1(*)C1(*))
	y29	C3(C3(OC)C2(C)C1(*))
	y30	C3(C3(OC)C2(C)C2(C))
	y31	C3(C3(OC)C3(CC)C1(*))
	y32	C3(C3(OC)C3(CC)C2(C))
	y33	C3(C3(OC)C3(CC)C3(CC))
Remaining C Atoms in R" Group	y34	C1(C2(C))
	y35	C1(C3(CC))
	y36	C2(C2(C)C1(*))
	y37	C2(C2(C)C2(C))
	y38	C2(C3(CC)C1(*))
	y39	C2(C3(CC)C2(C))
	y40	C2(C3(CC)C3(CC))
	y41	C3(C2(C)C1(*)C1(*))
	y42	C3(C2(C)C2(C)C1(*))
	y43	C3(C2(C)C2(C)C2(C))
	y44	C3(C3(CC)C1(*)C1(*))
	y45	C3(C3(CC)C2(C)C1(*))
	y46	C3(C3(CC)C2(C)C2(C))
	y47	C3(C3(CC)C3(CC)C1(*))
	y48	C3(C3(CC)C3(CC)C2(C))
	y49	C3(C3(CC)C3(CC)C3(CC))
O2 (Double Bonded)	y50	O2(=C4(OC))
C4 of Ester	y51	C4(=O2(*)O2(C)C1(*))
	y52	C4(=O2(*)O2(C)C2(C))
	y53	C4(=O2(*)O2(C)C3(CC))
C at Parent Level of C4 of Ester	y54	C1(C4(=OO))
	y55	C2(C4(=OO)C1(*))
	y56	C2(C4(=OO)C2(C))
	y57	C2(C4(=OO)C3(CC))
	y58	C3(C4(=OO)C1(*)C1(*))
	y59	C3(C4(=OO)C2(C)C1(*))
	y60	C3(C4(=OO)C2(C)C2(C))
	y61	C3(C4(=OO)C3(CC)C1(*))
	y62	C3(C4(=OO)C3(CC)C2(C))
	y63	C3(C4(=OO)C3(CC)C3(CC))
Remaining C Atoms in R Group	y64	C1(C2(C))

	y65	C1(C3(CC))
	y66	C2(C2(C)C1(*))
	y67	C2(C2(C)C2(C))
	y68	C2(C3(CC)C1(*))
	y69	C2(C3(CC)C2(C))
	y70	C2(C3(CC)C3(CC))
	y71	C3(C2(C)C1(*)C1(*))
	y72	C3(C2(C)C2(C)C1(*))
	y73	C3(C2(C)C2(C)C2(C))
	y74	C3(C3(CC)C1(*)C1(*))
	y75	C3(C3(CC)C2(C)C1(*))
	y76	C3(C3(CC)C2(C)C2(C))
	y77	C3(C3(CC)C3(CC)C1(*))
	y78	C3(C3(CC)C3(CC)C2(C))
	y79	C3(C3(CC)C3(CC)C3(CC))

Table 4.4: Signatures of Product Alcohol

ATOM TYPES	OCCURRENCE NOS. (zi)	SIGNATURES
O1	z1	O1(C1(*))
	z2	O1(C2(C))
	z3	O1(C3(CC))
C at Parent Level of O1	z4	C1(O1(*))
	z5	C2(O1(*)C1(*))
	z6	C2(O1(*)C2(C))
	z7	C2(O1(*)C3(CC))
	z8	C3(O1(*)C1(*)C1(*))
	z9	C3(O1(*)C2(C)C1(*))
	z10	C3(O1(*)C2(C)C2(C))
	z11	C3(O1(*)C3(CC)C1(*))
	z12	C3(O1(*)C3(CC)C2(C))
	z13	C3(O1(*)C3(CC)C3(CC))
C at Child Level of O1	z14	C1(C2(O))
	z15	C1(C3(OC))
	z16	C2(C2(O)C1(*))
	z17	C2(C2(O)C2(C))
	z18	C2(C2(O)C3(CC))
	z19	C2(C3(OC)C1(*))
	z20	C2(C3(OC)C2(C))
	z21	C2(C3(OC)C3(CC))

	z22	C3(C2(O)C1(*)C1(*))
	z23	C3(C2(O)C2(C)C1(*))
	z24	C3(C2(O)C2(C)C2(C))
	z25	C3(C2(O)C3(CC)C1(*))
	z26	C3(C2(O)C3(CC)C2(C))
	z27	C3(C2(O)C3(CC)C3(CC))
	z28	C3(C3(OC)C1(*)C1(*))
	z29	C3(C3(OC)C2(C)C1(*))
	z30	C3(C3(OC)C2(C)C2(C))
	z31	C3(C3(OC)C3(CC)C1(*))
	z32	C3(C3(OC)C3(CC)C2(C))
	z33	C3(C3(OC)C3(CC)C3(CC))
Remaining C Atoms in R' Group	z34	C1(C2(C))
	z35	C1(C3(CC))
	z36	C2(C2(C)C1(*))
	z37	C2(C2(C)C2(C))
	z38	C2(C3(CC)C1(*))
	z39	C2(C3(CC)C2(C))
	z40	C2(C3(CC)C3(CC))
	z41	C3(C2(C)C1(*)C1(*))
	z42	C3(C2(C)C2(C)C1(*))
	z43	C3(C2(C)C2(C)C2(C))
	z44	C3(C3(CC)C1(*)C1(*))
	z45	C3(C3(CC)C2(C)C1(*))
	z46	C3(C3(CC)C2(C)C2(C))
	z47	C3(C3(CC)C3(CC)C1(*))
	z48	C3(C3(CC)C3(CC)C2(C))
	z49	C3(C3(CC)C3(CC)C3(CC))

Table 4.5: Structures of designed Products and Reactants

Soln. No.	Reactants/ Products	Name	Objective Function Value

1	RCOOR´ R˝OH RCOOR˝ R´OH	ethyl 2,4,5,6-tetramethylheptanoate methanol methyl 2,4,5,6-tetramethylheptanoate ethanol	- - 341.81 K 282.81 K
2	RCOOR´ R˝OH RCOOR˝ R´OH	ethyl 2,4-dimethylpentanoate 3-methylbutan-2-ol 3-methylbutan-2-yl 2,4-dimethylpentanoate ethanol	- - 341.81 K 282.81 K
3	RCOOR´ R˝OH RCOOR˝ R´OH	ethyl 2,3-dimethylbutanoate 4-methylpentan-2-ol 4-methylpentan-2-yl 2,3-dimethylbutanoate ethanol	- - 341.81 K 282.81 K
4	RCOOR´ R˝OH RCOOR˝	ethyl 2,3-dimethylbutanoate 3-methylpentan-2-ol 3-methylpentan-2-yl 2,3-dimethylbutanoate	- - 341.81 K

	R´OH	ethanol	282.81 K
5	RCOOR´ R˝OH RCOOR˝ R´OH	ethyl 2-ethyl-3-isopropyl-4-methylpentanoate methanol methyl 2-ethyl-3-isopropyl-4-methylpentanoate ethanol	- - 341.81 K 282.81 K
6	RCOOR´ R˝OH RCOOR˝ R´OH	ethyl isobutyrate 4,5-dimethylhexan-2-ol 4,5-dimethylhexan-2-yl isobutyrate ethanol	- - 341.81 K 282.81 K
7	RCOOR´ R˝OH RCOOR˝ R´OH	ethyl 2-ethylpentanoate propanol propyl 2-ethylpentanoate ethanol	- - 341.41 K 282.81 K
8	RCOOR´ R˝OH	ethyl 2-methylhexanoate propanol	- -

	RCOOR˝	propyl 2-methylhexanoate	341.41 K
	R´OH	ethanol	282.81 K
9	RCOOR´	ethyl 2-propylhexanoate	-
	R˝OH	methanol	-
	RCOOR˝	methyl 2-propylhexanoate	341.41 K
	R´OH	ethanol	282.81 K
10	RCOOR´	ethyl isobutyrate	-
	R˝OH	5-methylhexan-3-ol	-
	RCOOR˝	5-methylhexan-3-yl isobutyrate	337.87 K
	R´OH	ethanol	282.81 K
11	RCOOR´	ethyl 2-methylbutanoate	-
	R˝OH	2-methylpentan-3-ol	-
	RCOOR˝	2-methylpentan-3-yl 2-methylbutanoate	337.87 K
	R´OH	ethanol	282.81 K

4.2. Case Study 2

Like in the first case study, we rely on the transesterification reaction portrayed in Fig. 3.2. We use the same terminologies as those in section 4.1. The aim of this case study is to design RCOOR´, R˝OH, RCOOR˝ and R´OH such that the standard Gibbs free energy change of the transesterification reaction in gas phase, ΔG^{o}_{rxn}, is minimized. Also, RCOOR´ and R˝OH are constrained by log(*P*) and log(LC_{50}) respectively. RCOOR˝ and R´OH are subjected to their respective set of T_b and F_p constraints. The property ranges are listed in Table 4.6. ΔG^{0}_{rxn} has been evaluated using the following definition:

$$\Delta G^{0}_{rxn} = \sum a_{product} G_{f,product} - \sum b_{reactant} G_{f,reactant} \tag{4.1}$$

Where, $G_{f,\ product}$ and $G_{f,\ reactant}$ are standard Gibbs energy of formation values and $a_{product}$ and $b_{reactant}$ are stoichiometric coefficients of each product and reactant respectively. The free energy of formation in gas phase (G_f) in kJ/mol of each reactant and product was expressed in terms of their structures using the following GCM developed by Hukkerikar *et al.* (2012):

$$G_f - G_{f0} = \sum_i N_i C_i + \sum_j M_j D_j + \sum_k E_k O_k \tag{4.2}$$

The GCM in Eq. (4.2) has the same form as the GCMs mentioned in Table 4.2. The other utilized property models are same as those listed in Table 4.2.

Table 4.6: Property Constraints on Reactants and Products

Reactant/Product	Property	Upper Bound	Lower Bound
All	ΔG^{o}_{rxn} (kJ/mol)	Minimum	
RCOOR´	log(*P*)	5	-
R˝OH	log(LC_{50})	-	-1
RCOOR˝	Boiling Point (K) Flash Point (K)	485 375	395 295
R´OH	Boiling Point (K) Flash Point (K)	470 380	385 300

The strategy in section 3.3, to reduce the number of signatures generated, is relevant to this case study. This is because the highest height required is 3 in this case, owing to some second order groups in the GCMs. All other groups in the GCMs and TIs can be expressed using height 2 signatures. An MINLP problem, in accordance with section 3.2.3, of minimizing the standard Gibbs energy change of transesterification reaction in gas phase, was set up in GAMS. Additionally, the structural and property constraints of the reactant and product esters and

alcohols along with RABS relationships were included taking into consideration the above strategy. The structural constraints owing to the height disparity, mentioned in section 3.3, are also included in the MINLP problem. The MINLP problem was solved using the DICOPT solver. Also, R, R′ and R″ are assumed to be homogeneous, acyclic, and saturated groups. Also, the degree of atoms in R, R′ and R″ do not exceed 3. Unlike the first case study, we have not ignored the second order groups in the GCMs. Thus, the GCMs have not been approximated. Also, no third order groups exist for our case study. Taking into account the aforementioned assumptions, the signatures of atoms of reactant ester and reactant alcohol are listed in Table 4.7 and Table 4.8. The signatures of products, not listed here, are same as that of the reactants. They have different occurrence no. variables associated with them, however. The obtained results corresponding to the optimal value of the Gibbs free energy change of the reaction have been presented in Table 4.9. It is worth mentioning that although we have optimized the Std. Gibbs Energy change in gas phase, one can definitely use the methodology described in section 3.2.3 to carry out the optimization in other phases as long as the suitable thermodynamic models are available. In aqueous phase, for example, the Gibbs Energy of formation computation can be carried out using the GCM developed by Jankowski et al. (2008).

Table 4.7: Signatures of Reactant Ester

ATOM TYPES	OCCURRENCE NOS. (wi)	SIGNATURES
O2 (Single Bonded)	w1 w2 w3	O2(C1(*)C4(=OC)) O2(C2(C)C4(=OC)) O2(C3(CC)C4(=OC))
C at Parent Level of O2	w4	C1(O2(C))

	w5	C2(O2(C)C1(*))
	w6	C2(O2(C)C2(C))
	w7	C2(O2(C)C3(CC))
	w8	C3(O2(C)C1(*)C1(*))
	w9	C3(O2(C)C2(C)C1(*))
	w10	C3(O2(C)C2(C)C2(C))
	w11	C3(O2(C)C3(CC)C1(*))
	w12	C3(O2(C)C3(CC)C2(C))
	w13	C3(O2(C)C3(CC)C3(CC))
	w14	C3(C1(*)O2(C4(=OC))C3(C1(*)C1(*)))
	w15	C3(C1(*)O2(C4(=OC))C3(C1(*)C2(C)))
	w16	C3(C1(*)O2(C4(=OC))C3(C1(*)C3(CC)))
C at Child Level of O2	w17	C1(C2(O))
	w18	C1(C3(OC))
	w19	C2(C2(O)C1(*))
	w20	C2(C2(O)C2(C))
	w21	C2(C2(O)C3(CC))
	w22	C2(C3(OC)C1(*))
	w23	C2(C3(OC)C2(C))
	w24	C2(C3(OC)C3(CC))
	w25	C3(C2(O)C1(*)C1(*))
	w26	C3(C2(O)C2(C)C1(*))
	w27	C3(C2(O)C2(C)C2(C))
	w28	C3(C2(O)C3(CC)C1(*))
	w29	C3(C2(O)C3(CC)C2(C))
	w30	C3(C2(O)C3(CC)C3(CC))
	w31	C3(C3(OC)C1(*)C1(*))
	w32	C3(C3(OC)C2(C)C1(*))
	w33	C3(C3(OC)C2(C)C2(C))
	w34	C3(C3(OC)C3(CC)C1(*))
	w35	C3(C3(OC)C3(CC)C2(C))
	w36	C3(C3(OC)C3(CC)C3(CC))
	w37	C3(C1(*)C1(*)C3(C1(*)O2(C)))
	w38	C3(C1(*)C2(C1(*))C3(C1(*)O2(C)))
	w39	C3(C1(*)C2(C2(C))C3(C1(*)O2(C)))
	w40	C3(C1(*)C2(C3(CC))C3(C1(*)O2(C)))
	w41	C3(C1(*)C3(C1(*)C1(*))C3(C1(*)O2(C)))
	w42	C3(C1(*)C3(C2(C)C1(*))C3(C1(*)O2(C)))
	w43	C3(C1(*)C3(C2(C)C2(C))C3(C1(*)O2(C)))
	w44	C3(C1(*)C3(C3(CC)C1(*))C3(C1(*)O2(C)))
	w45	C3(C1(*)C3(C3(CC)C2(C))C3(C1(*)O2(C)))
	w46	C3(C1(*)C3(C3(CC)C3(CC))C3(C1(*)O2(C)))

	w47	C3(C1(*)C2(O2(C))C3(C1(*)C1(*)))
	w48	C3(C1(*)C2(O2(C))C3(C1(*)C2(C)))
	w49	C3(C1(*)C2(O2(C))C3(C1(*)C3(CC)))
	w50	C3(C1(*)C3(O2(C)C2(C))C3(C1(*)C1(*)))
	w51	C3(C1(*)C3(O2(C)C2(C))C3(C1(*)C2(C)))
	w52	C3(C1(*)C3(O2(C)C2(C))C3(C1(*)C3(CC)))
	w53	C3(C1(*)C3(O2(C)C3(CC))C3(C1(*)C1(*)))
	w54	C3(C1(*)C3(O2(C)C3(CC))C3(C1(*)C2(C)))
	w55	C3(C1(*)C3(O2(C)C3(CC))C3(C1(*)C3(CC)))
C at Height 3 from O2	w56	C3(C1(*)C1(*)C3(C1(*)C2(O)))
	w57	C3(C1(*)C2(C1(*))C3(C1(*)C2(O)))
	w58	C3(C1(*)C2(C2(C))C3(C1(*)C2(O)))
	w59	C3(C1(*)C2(C3(CC))C3(C1(*)C2(O)))
	w60	C3(C1(*)C3(C1(*)C1(*))C3(C1(*)C2(O)))
	w61	C3(C1(*)C3(C2(C)C1(*))C3(C1(*)C2(O)))
	w62	C3(C1(*)C3(C2(C)C2(C))C3(C1(*)C2(O)))
	w63	C3(C1(*)C3(C3(CC)C1(*))C3(C1(*)C2(O)))
	w64	C3(C1(*)C3(C3(CC)C2(C))C3(C1(*)C2(O)))
	w65	C3(C1(*)C3(C3(CC)C3(CC))C3(C1(*)C2(O)))
	w66	C3(C1(*)C1(*)C3(C1(*)C3(OC)))
	w67	C3(C1(*)C2(C1(*))C3(C1(*)C3(OC)))
	w68	C3(C1(*)C2(C2(C))C3(C1(*)C3(OC)))
	w69	C3(C1(*)C2(C3(CC))C3(C1(*)C3(OC)))
	w70	C3(C1(*)C3(C1(*)C1(*))C3(C1(*)C3(OC)))
	w71	C3(C1(*)C3(C2(C)C1(*))C3(C1(*)C3(OC)))
	w72	C3(C1(*)C3(C2(C)C2(C))C3(C1(*)C3(OC)))
	w73	C3(C1(*)C3(C3(CC)C1(*))C3(C1(*)C3(OC)))
	w74	C3(C1(*)C3(C3(CC)C2(C))C3(C1(*)C3(OC)))
	w75	C3(C1(*)C3(C3(CC)C3(CC))C3(C1(*)C3(OC)))
	w76	C3(C1(*)C2(C2(O))C3(C1(*)C1(*)))
	w77	C3(C1(*)C2(C2(O))C3(C1(*)C2(C)))
	w78	C3(C1(*)C2(C2(O))C3(C1(*)C3(CC)))
	w79	C3(C1(*)C2(C3(OC))C3(C1(*)C1(*)))
	w80	C3(C1(*)C2(C3(OC))C3(C1(*)C2(C)))
	w81	C3(C1(*)C2(C3(OC))C3(C1(*)C3(CC)))
	w82	C3(C1(*)C3(C2(O)C2(C))C3(C1(*)C1(*)))
	w83	C3(C1(*)C3(C2(O)C2(C))C3(C1(*)C2(C)))
	w84	C3(C1(*)C3(C2(O)C2(C))C3(C1(*)C3(CC)))
	w85	C3(C1(*)C3(C2(O)C3(CC))C3(C1(*)C1(*)))
	w86	C3(C1(*)C3(C2(O)C3(CC))C3(C1(*)C2(C)))
	w87	C3(C1(*)C3(C2(O)C3(CC))C3(C1(*)C3(CC)))
	w88	C3(C1(*)C3(C3(OC)C2(C))C3(C1(*)C1(*)))

	w89	C3(C1(*)C3(C3(OC)C2(C))C3(C1(*)C2(C)))
	w90	C3(C1(*)C3(C3(OC)C2(C))C3(C1(*)C3(CC)))
	w91	C3(C1(*)C3(C3(OC)C3(CC))C3(C1(*)C1(*)))
	w92	C3(C1(*)C3(C3(OC)C3(CC))C3(C1(*)C2(C)))
	w93	C3(C1(*)C3(C3(OC)C3(CC))C3(C1(*)C3(CC)))
Remaining C atoms in R' Group	w94	C1(C2(C))
	w95	C1(C3(CC))
	w96	C2(C2(C)C1(*))
	w97	C2(C2(C)C2(C))
	w98	C2(C3(CC)C1(*))
	w99	C2(C3(CC)C2(C))
	w100	C2(C3(CC)C3(CC))
	w101	C3(C2(C)C1(*)C1(*))
	w102	C3(C2(C)C2(C)C1(*))
	w103	C3(C2(C)C2(C)C2(C))
	w104	C3(C3(CC)C1(*)C1(*))
	w105	C3(C3(CC)C2(C)C1(*))
	w106	C3(C3(CC)C2(C)C2(C))
	w107	C3(C3(CC)C3(CC)C1(*))
	w108	C3(C3(CC)C3(CC)C2(C))
	w109	C3(C3(CC)C3(CC)C3(CC))
	w110	C3(C1(*)C1(*)C3(C1(*)C2(C)))
	w111	C3(C1(*)C1(*)C3(C1(*)C3(CC)))
	w112	C3(C1(*)C2(C1(*))C3(C1(*)C2(C)))
	w113	C3(C1(*)C2(C1(*))C3(C1(*)C3(CC)))
	w114	C3(C1(*)C2(C2(C))C3(C1(*)C1(*)))
	w115	C3(C1(*)C2(C2(C))C3(C1(*)C2(C)))
	w116	C3(C1(*)C2(C2(C))C3(C1(*)C3(CC)))
	w117	C3(C1(*)C2(C3(CC))C3(C1(*)C1(*)))
	w118	C3(C1(*)C2(C3(CC))C3(C1(*)C2(C)))
	w119	C3(C1(*)C2(C3(CC))C3(C1(*)C3(CC)))
	w120	C3(C1(*)C3(C1(*)C1(*))C3(C1(*)C2(C)))
	w121	C3(C1(*)C3(C1(*)C1(*))C3(C1(*)C3(CC)))
	w122	C3(C1(*)C3(C2(C)C1(*))C3(C1(*)C2(C)))
	w123	C3(C1(*)C3(C2(C)C1(*))C3(C1(*)C3(CC)))
	w124	C3(C1(*)C3(C2(C)C2(C))C3(C1(*)C1(*)))
	w125	C3(C1(*)C3(C2(C)C2(C))C3(C1(*)C2(C)))
	w126	C3(C1(*)C3(C2(C)C2(C))C3(C1(*)C3(CC)))
	w127	C3(C1(*)C3(C3(CC)C1(*))C3(C1(*)C3(CC)))
	w128	C3(C1(*)C3(C3(CC)C2(C))C3(C1(*)C1(*)))
	w129	C3(C1(*)C3(C3(CC)C2(C))C3(C1(*)C2(C)))

	w130	C3(C1(*)C3(C3(CC)C2(C))C3(C1(*)C3(CC)))
	w131	C3(C1(*)C3(C3(CC)C3(CC))C3(C1(*)C1(*)))
	w132	C3(C1(*)C3(C3(CC)C3(CC))C3(C1(*)C2(C)))
	w133	C3(C1(*)C3(C3(CC)C3(CC))C3(C1(*)C3(CC)))
O2 (Double Bonded)	w134	O2(=C4(OC))
C4 of Ester Group	w135	C4(=O2(*)O2(C)C1(*))
	w136	C4(=O2(*)O2(C3(CC))C1(*))
	w137	C4(=O2(*)O2(C)C2(C))
	w138	C4(=O2(*)O2(C)C3(CC))
C at Parent Level of C4	w139	C1(C4(=OO))
	w140	C2(C4(=OO)C1(*))
	w141	C2(C4(=OO)C2(C))
	w142	C2(C4(=OO)C3(CC))
	w143	C3(C4(=OO)C1(*)C1(*))
	w144	C3(C4(=OO)C2(C)C1(*))
	w145	C3(C4(=OO)C2(C)C2(C))
	w146	C3(C4(=OO)C3(CC)C1(*))
	w147	C3(C4(=OO)C3(CC)C2(C))
	w148	C3(C4(=OO)C3(CC)C3(CC))
	w149	C3(C1(*)C4(=O2(*)O2(C))C3(C1(*)C1(*)))
	w150	C3(C1(*)C4(=O2(*)O2(C))C3(C1(*)C2(C)))
	w151	C3(C1(*)C4(=O2(*)O2(C))C3(C1(*)C3(CC)))
C at Child Level of C4	w152	C3(C1(*)C1(*)C3(C1(*)C4(=OO)))
	w153	C3(C1(*)C2(C1(*))C3(C1(*)C4(=OO)))
	w154	C3(C1(*)C2(C2(C))C3(C1(*)C4(=OO)))
	w155	C3(C1(*)C2(C3(CC))C3(C1(*)C4(=OO)))
	w156	C3(C1(*)C3(C1(*)C1(*))C3(C1(*)C4(=OO)))
	w157	C3(C1(*)C3(C2(C)C1(*))C3(C1(*)C4(=OO)))
	w158	C3(C1(*)C3(C2(C)C2(C))C3(C1(*)C4(=OO)))
	w159	C3(C1(*)C3(C3(CC)C1(*))C3(C1(*)C4(=OO)))
	w160	C3(C1(*)C3(C3(CC)C2(C))C3(C1(*)C4(=OO)))
	w161	C3(C1(*)C3(C3(CC)C3(CC))C3(C1(*)C4(=OO)))
	w162	C3(C1(*)C2(C4(=OO))C3(C1(*)C1(*)))
	w163	C3(C1(*)C2(C4(=OO))C3(C1(*)C2(C)))
	w164	C3(C1(*)C2(C4(=OO))C3(C1(*)C3(CC)))
	w165	C3(C1(*)C3(C4(=OO)C2(C))C3(C1(*)C1(*)))
	w166	C3(C1(*)C3(C4(=OO)C3(CC))C3(C1(*)C1(*)))
	w167	C3(C1(*)C3(C4(=OO)C2(C))C3(C1(*)C2(C)))
	w168	C3(C1(*)C3(C4(=OO)C3(CC))C3(C1(*)C2(C)))
	w169	C3(C1(*)C3(C4(=OO)C2(C))C3(C1(*)C3(CC)))
	w170	C3(C1(*)C3(C4(=OO)C3(CC))C3(C1(*)C3(CC)))

Remaining C Atoms in R Group	w171	C1(C2(C))
	w172	C1(C3(CC))
	w173	C2(C2(C)C1(*))
	w174	C2(C2(C)C2(C))
	w175	C2(C3(CC)C1(*))
	w176	C2(C3(CC)C2(C))
	w177	C2(C3(CC)C3(CC))
	w178	C3(C2(C)C1(*)C1(*))
	w179	C3(C2(C)C2(C)C1(*))
	w180	C3(C2(C)C2(C)C2(C))
	w181	C3(C3(CC)C1(*)C1(*))
	w182	C3(C3(CC)C2(C)C1(*))
	w183	C3(C3(CC)C2(C)C2(C))
	w184	C3(C3(CC)C3(CC)C1(*))
	w185	C3(C3(CC)C3(CC)C2(C))
	w186	C3(C3(CC)C3(CC)C3(CC))
	w187	C3(C1(*)C1(*))C3(C1(*)C2(C)))
	w188	C3(C1(*)C1(*))C3(C1(*)C3(CC)))
	w189	C3(C1(*)C2(C1(*))C3(C1(*)C2(C)))
	w190	C3(C1(*)C2(C1(*))C3(C1(*)C3(CC)))
	w191	C3(C1(*)C2(C2(C))C3(C1(*)C1(*)))
	w192	C3(C1(*)C2(C2(C))C3(C1(*)C2(C)))
	w193	C3(C1(*)C2(C2(C))C3(C1(*)C3(CC)))
	w194	C3(C1(*)C2(C3(CC))C3(C1(*)C1(*)))
	w195	C3(C1(*)C2(C3(CC))C3(C1(*)C2(C)))
	w196	C3(C1(*)C2(C3(CC))C3(C1(*)C3(CC)))
	w197	C3(C1(*)C3(C1(*)C1(*))C3(C1(*)C2(C)))
	w198	C3(C1(*)C3(C1(*)C1(*))C3(C1(*)C3(CC)))
	w199	C3(C1(*)C3(C2(C)C1(*))C3(C1(*)C2(C)))
	w200	C3(C1(*)C3(C2(C)C1(*))C3(C1(*)C3(CC)))
	w201	C3(C1(*)C3(C2(C)C2(C))C3(C1(*)C1(*)))
	w202	C3(C1(*)C3(C2(C)C2(C))C3(C1(*)C2(C)))
	w203	C3(C1(*)C3(C2(C)C2(C))C3(C1(*)C3(CC)))
	w204	C3(C1(*)C3(C3(CC)C1(*))C3(C1(*)C3(CC)))
	w205	C3(C1(*)C3(C3(CC)C2(C))C3(C1(*)C1(*)))
	w206	C3(C1(*)C3(C3(CC)C2(C))C3(C1(*)C2(C)))
	w207	C3(C1(*)C3(C3(CC)C2(C))C3(C1(*)C3(CC)))
	w208	C3(C1(*)C3(C3(CC)C3(CC))C3(C1(*)C1(*)))
	w209	C3(C1(*)C3(C3(CC)C3(CC))C3(C1(*)C2(C)))
	w210	C3(C1(*)C3(C3(CC)C3(CC))C3(C1(*)C3(CC)))

Table 4.8: Signatures of Reactant Alcohol

ATOM TYPES	OCCURRENCE NOS. (xi)	SIGNATURES
O1	x1	O1(C1(*))
	x2	O1(C2(C))
	x3	O1(C3(CC))
C at Parent Level of O1	x4	C1(O1(*))
	x5	C2(O1(*)C1(*))
	x6	C2(O1(*)C2(C))
	x7	C2(O1(*)C3(CC))
	x8	C3(O1(*)C1(*)C1(*))
	x9	C3(O1(*)C2(C)C1(*))
	x10	C3(O1(*)C2(C)C2(C))
	x11	C3(O1(*)C3(CC)C1(*))
	x12	C3(O1(*)C3(CC)C2(C))
	x13	C3(O1(*)C3(CC)C3(CC))
	x14	C3(C1(*)O1(*)C3(C1(*)C1(*)))
	x15	C3(C1(*)O1(*)C3(C1(*)C2(C)))
	x16	C3(C1(*)O1(*)C3(C1(*)C3(CC)))
C at Child Level of O1	x17	C1(C2(O))
	x18	C1(C3(OC))
	x19	C2(C2(O)C1(*))
	x20	C2(C2(O)C2(C))
	x21	C2(C2(O)C3(CC))
	x22	C2(C3(OC)C1(*))
	x23	C2(C3(OC)C2(C))
	x24	C2(C3(OC)C3(CC))
	x25	C3(C2(O)C1(*)C1(*))
	x26	C3(C2(O)C2(C)C1(*))
	x27	C3(C2(O)C2(C)C2(C))
	x28	C3(C2(O)C3(CC)C1(*))
	x29	C3(C2(O)C3(CC)C2(C))
	x30	C3(C2(O)C3(CC)C3(CC))
	x31	C3(C3(OC)C1(*)C1(*))
	x32	C3(C3(OC)C2(C)C1(*))
	x33	C3(C3(OC)C2(C)C2(C))
	x34	C3(C3(OC)C3(CC)C1(*))
	x35	C3(C3(OC)C3(CC)C2(C))
	x36	C3(C3(OC)C3(CC)C3(CC))
	x37	C3(C1(*)C1(*)C3(C1(*)O1(*)))
	x38	C3(C1(*)C2(C1(*))C3(C1(*)O1(*)))
	x39	C3(C1(*)C2(C2(C))C3(C1(*)O1(*)))

	x40	C3(C1(*)C2(C3(CC))C3(C1(*)O1(*)))
	x41	C3(C1(*)C3(C1(*)C1(*))C3(C1(*)O1(*)))
	x42	C3(C1(*)C3(C2(C)C1(*))C3(C1(*)O1(*)))
	x43	C3(C1(*)C3(C2(C)C2(C))C3(C1(*)O1(*)))
	x44	C3(C1(*)C3(C3(CC)C1(*))C3(C1(*)O1(*)))
	x45	C3(C1(*)C3(C3(CC)C2(C))C3(C1(*)O1(*)))
	x46	C3(C1(*)C3(C3(CC)C3(CC))C3(C1(*)O1(*)))
	x47	C3(C1(*)C2(O1(*))C3(C1(*)C1(*)))
	x48	C3(C1(*)C2(O1(*))C3(C1(*)C2(C)))
	x49	C3(C1(*)C2(O1(*))C3(C1(*)C3(CC)))
	x50	C3(C1(*)C3(O1(*)C2(C))C3(C1(*)C1(*)))
	x51	C3(C1(*)C3(O1(*)C2(C))C3(C1(*)C2(C)))
	x52	C3(C1(*)C3(O1(*)C2(C))C3(C1(*)C3(CC)))
	x53	C3(C1(*)C3(O1(*)C3(CC))C3(C1(*)C1(*)))
	x54	C3(C1(*)C3(O1(*)C3(CC))C3(C1(*)C2(C)))
	x55	C3(C1(*)C3(O1(*)C3(CC))C3(C1(*)C3(CC)))
C at Height 3 from O1	x56	C3(C1(*)C1(*)C3(C1(*)C2(O)))
	x57	C3(C1(*)C2(C1(*))C3(C1(*)C2(O)))
	x58	C3(C1(*)C2(C2(C))C3(C1(*)C2(O)))
	x59	C3(C1(*)C2(C3(CC))C3(C1(*)C2(O)))
	x60	C3(C1(*)C3(C1(*)C1(*))C3(C1(*)C2(O)))
	x61	C3(C1(*)C3(C2(C)C1(*))C3(C1(*)C2(O)))
	x62	C3(C1(*)C3(C2(C)C2(C))C3(C1(*)C2(O)))
	x63	C3(C1(*)C3(C3(CC)C1(*))C3(C1(*)C2(O)))
	x64	C3(C1(*)C3(C3(CC)C2(C))C3(C1(*)C2(O)))
	x65	C3(C1(*)C3(C3(CC)C3(CC))C3(C1(*)C2(O)))
	x66	C3(C1(*)C1(*)C3(C1(*)C3(OC)))
	x67	C3(C1(*)C2(C1(*))C3(C1(*)C3(OC)))
	x68	C3(C1(*)C2(C2(C))C3(C1(*)C3(OC)))
	x69	C3(C1(*)C2(C3(CC))C3(C1(*)C3(OC)))
	x70	C3(C1(*)C3(C1(*)C1(*))C3(C1(*)C3(OC)))
	x71	C3(C1(*)C3(C2(C)C1(*))C3(C1(*)C3(OC)))
	x72	C3(C1(*)C3(C2(C)C2(C))C3(C1(*)C3(OC)))
	x73	C3(C1(*)C3(C3(CC)C1(*))C3(C1(*)C3(OC)))
	x74	C3(C1(*)C3(C3(CC)C2(C))C3(C1(*)C3(OC)))
	x75	C3(C1(*)C3(C3(CC)C3(CC))C3(C1(*)C3(OC)))
	x76	C3(C1(*)C2(C2(O))C3(C1(*)C1(*)))
	x77	C3(C1(*)C2(C2(O))C3(C1(*)C2(C)))
	x78	C3(C1(*)C2(C2(O))C3(C1(*)C3(CC)))
	x79	C3(C1(*)C2(C3(OC))C3(C1(*)C1(*)))
	x80	C3(C1(*)C2(C3(OC))C3(C1(*)C2(C)))
	x81	C3(C1(*)C2(C3(OC))C3(C1(*)C3(CC)))
	x82	C3(C1(*)C3(C2(O)C2(C))C3(C1(*)C1(*)))

	x83	C3(C1(*)C3(C2(O)C2(C))C3(C1(*)C2(C)))
	x84	C3(C1(*)C3(C2(O)C2(C))C3(C1(*)C3(CC)))
	x85	C3(C1(*)C3(C2(O)C3(CC))C3(C1(*)C1(*)))
	x86	C3(C1(*)C3(C2(O)C3(CC))C3(C1(*)C2(C)))
	x87	C3(C1(*)C3(C2(O)C3(CC))C3(C1(*)C3(CC)))
	x88	C3(C1(*)C3(C3(OC)C2(C))C3(C1(*)C1(*)))
	x89	C3(C1(*)C3(C3(OC)C2(C))C3(C1(*)C2(C)))
	x90	C3(C1(*)C3(C3(OC)C2(C))C3(C1(*)C3(CC)))
	x91	C3(C1(*)C3(C3(OC)C3(CC))C3(C1(*)C1(*)))
	x92	C3(C1(*)C3(C3(OC)C3(CC))C3(C1(*)C2(C)))
	x93	C3(C1(*)C3(C3(OC)C3(CC))C3(C1(*)C3(CC)))
Remaining C atoms in R"	x94	C1(C2(C))
Group	x95	C1(C3(CC))
	x96	C2(C2(C)C1(*))
	x97	C2(C2(C)C2(C))
	x98	C2(C3(CC)C1(*))
	x99	C2(C3(CC)C2(C))
	x100	C2(C3(CC)C3(CC))
	x101	C3(C2(C)C1(*)C1(*))
	x102	C3(C2(C)C2(C)C1(*))
	x103	C3(C2(C)C2(C)C2(C))
	x104	C3(C3(CC)C1(*)C1(*))
	x105	C3(C3(CC)C2(C)C1(*))
	x106	C3(C3(CC)C2(C)C2(C))
	x107	C3(C3(CC)C3(CC)C1(*))
	x108	C3(C3(CC)C3(CC)C2(C))
	x109	C3(C3(CC)C3(CC)C3(CC))
	x110	C3(C1(*)C1(*)C3(C1(*)C2(C)))
	x111	C3(C1(*)C1(*)C3(C1(*)C3(CC)))
	x112	C3(C1(*)C2(C1(*))C3(C1(*)C2(C)))
	x113	C3(C1(*)C2(C1(*))C3(C1(*)C3(CC)))
	x114	C3(C1(*)C2(C2(C))C3(C1(*)C1(*)))
	x115	C3(C1(*)C2(C2(C))C3(C1(*)C2(C)))
	x116	C3(C1(*)C2(C2(C))C3(C1(*)C3(CC)))
	x117	C3(C1(*)C2(C3(CC))C3(C1(*)C1(*)))
	x118	C3(C1(*)C2(C3(CC))C3(C1(*)C2(C)))
	x119	C3(C1(*)C2(C3(CC))C3(C1(*)C3(CC)))
	x120	C3(C1(*)C3(C1(*)C1(*))C3(C1(*)C2(C)))
	x121	C3(C1(*)C3(C1(*)C1(*))C3(C1(*)C3(CC)))
	x122	C3(C1(*)C3(C2(C)C1(*))C3(C1(*)C2(C)))
	x123	C3(C1(*)C3(C2(C)C1(*))C3(C1(*)C3(CC)))
	x124	C3(C1(*)C3(C2(C)C2(C))C3(C1(*)C1(*)))

	x125	C3(C1(*)C3(C2(C)C2(C))C3(C1(*)C2(C)))
	x126	C3(C1(*)C3(C2(C)C2(C))C3(C1(*)C3(CC)))
	x127	C3(C1(*)C3(C3(CC)C1(*))C3(C1(*)C3(CC)))
	x128	C3(C1(*)C3(C3(CC)C2(C))C3(C1(*)C1(*)))
	x129	C3(C1(*)C3(C3(CC)C2(C))C3(C1(*)C2(C)))
	x130	C3(C1(*)C3(C3(CC)C2(C))C3(C1(*)C3(CC)))
	x131	C3(C1(*)C3(C3(CC)C3(CC))C3(C1(*)C1(*)))
	x132	C3(C1(*)C3(C3(CC)C3(CC))C3(C1(*)C2(C)))
	x133	C3(C1(*)C3(C3(CC)C3(CC))C3(C1(*)C3(CC)))

Table 4.9: Structures of designed Reactants and Products

Soln. No.	Reactants/ Products		Name	Objective Function Value
1	RCOOR´ R˝OH RCOOR˝ R´OH		2-methylpentan-3-yl pentanoate ethanol Ethyl pentanoate 2-methylpentan-3-ol	-13.62 kJ/mol
2	RCOOR´ R˝OH		pentan-2-yl 3-ethylpentanoate ethanol	-13.62 kJ/mol

	RCOOR˝ R´OH		ethyl 3-ethylpentanoate pentan-2-ol	
3	RCOOR´ R˝OH RCOOR˝ R´OH		pentan-3-yl 2,3-dimethylbutanoate methanol methyl 2,3-dimethylbutanoate pentan-3-ol	-13.62 kJ/mol
4	RCOOR´ R˝OH RCOOR˝ R´OH		hexan-3-yl 2,3-dimethylbutanoate 3-methylpentan-2-ol 3-methylpentan-2-yl 2,3-dimethylbutanoate hexan-3-ol	-13.62 kJ/mol
5	RCOOR´ R˝OH RCOOR˝		pentan-3-yl 3,4-dimethylpentanoate methanol methyl 3,4-dimethylpentanoate	-13.62 kJ/mol

	R´OH		pentan-3-ol	
6	RCOOR´ R˝OH RCOOR˝ R´OH		pentan-3-yl 2,3-dimethylpentanoate methanol methyl 2,3-dimethylpentanoate pentan-3-ol	-13.62 kJ/mol
7	RCOOR´ R˝OH RCOOR˝ R´OH		pentan-2-yl 2-ethylbutanoate methanol methyl 2-ethylbutanoate pentan-2-ol	-13.62 kJ/mol

4.3. Case Study 3

Once again, we use the transesterification reaction portrayed in Fig. 3.2 to demonstrate our algorithm. The aim of this case study is to design RCOOR´ and R˝OH with respective optimal log(LC_{50}) values and RCOOR˝ and R´OH with respective optimal F_p. The F_p of the products and log(LC_{50}) of the reactants are to be maximized. Additionally, T_b of the reactants are constrained

and T_b and melting point (T_m) of the products are constrained. The lower and upper bounds on T_b and T_m are mentioned in Table 4.10. The *log(LC_{50})* value of RCOOR′ and R″OH decreases with the increase in the values of occurrence numbers of signatures of atoms in R, R′ and R″. On the other hand, F_p value of RCOOR″ and R′OH increases with the rise in values of occurrence numbers of signatures of atoms in R, R′ and R″. This case is similar to that of the example presented in section 3.2.4 where A is the RCO group, OR′ is B and R″O group is C. Since, in accordance with section 3.2.4, the property objective functions are conflicting, there is not going to be one single optima value for each function that will capture the various trade-offs. Thus, a pareto optimal set will have to be generated. The log(LC_{50}) computation is carried out using the following GCM developed by Martin and Young (2001):

$$-\log(LC_{50}) = \sum_{i=1}^{ng} n_i \alpha_i \tag{4.3}$$

Where, n_i is the number of groups of type *i* in the compound, α_i is the toxicity contribution of group *i* and *ng* is the number of groups in the model. T_m computation in Kelvin is carried out using the following GCM developed by Hukkerikar et al. (2012):

$$\exp(T_m/T_{m0}) = \sum_i N_i C_i + \sum_j M_j D_j + \sum_k E_k O_k \quad (4.4)$$

The T_m GCM has the same form as the GCMs in Table 4.2. Also, for computation of T_b and F_p, the GCMs mentioned in Table 4.2 are utilized.

Table 4.10: Property Constraints on Reactants and Products

Reactants/ Products	Property	Upper Bound	Lower Bound
RCOOR´	log(LC_{50})	Maximum	
	Boiling Point (K)	430	350
R˝OH	log(LC_{50})	Maximum	
	Boiling Point (K)	430	360
RCOOR˝	Flash Point (K)	Maximum	
	Boiling Point (K)	485	395
	Melting Point (K)	280	200
R´OH	Flash Point (K)	Maximum	

	Boiling Point (K)	470	385
	Melting Point (K)	275	190

To make the T_b and T_m GCMs linear, we replace $exp(T_b/T_{b0})$ and $exp(T_m/T_{m0})$ by some variables U and V. The constraints on T_m and T_b are also accordingly modified so that they then become constraints on U and V. Since all the models now become linear, we will set up an MOMILP problem. In accordance with section 3.2.4, the MOMILP problem consists of each of the objective functions, property constraints, structural feasibility constraints and RABS relationships. Like in section 4.1, we have ignored the second order groups. There are no third order groups for our case. R, R´ and R˝ are assumed to be homogeneous, acyclic, and saturated groups. Also, the degree of atoms in R, R´ and R˝ do not exceed 3. The signatures of atoms in reactants and products are same as those listed in Table 4.3 and Table 4.4. The signatures of reactant alcohol and product alcohol have different occurrence no. variable associated with them, however. Similar is the case for the reactant and product ester. To solve the MOMILP problem, we utilized AUGMECON. In the AUGMECON scheme, the ensuing MILP problems were solved using CPLEX in GAMS. We chose the number of grid points to be 5. Choosing additional grid points may provide additional pareto optimal solutions. We have listed the pareto optimal solutions obtained for our case study in Table 4.11.

Table 4.11: Pareto Optimal Solutions

Solution No.	Reactants/ Products	Name	Objective Function Value
1	RCOOR´	butyl acetate	-3.52
	R˝OH	pentanol	-2.19
	RCOOR˝	pentyl acetate	160.24 K
	R´OH	butanol	154.98 K
2	RCOOR´	pentan-2-yl acetate	-3.84
	R˝OH	pentanol	-2.19
	RCOOR˝	pentyl acetate	160.24 K
	R´OH	pentan-2-ol	158.92 K
3	RCOOR´	2-methylbutyl acetate	-3.84
	R˝OH	butanol	-1.74
	RCOOR˝	butyl acetate	148.83 K
	R´OH	2-methylbutanol	158.92 K
4	RCOOR´	pentan-3-yl acetate	-3.84

	R˝OH RCOOR˝ R´OH	Pentanol pentyl acetate pentan-3-ol	-2.19 160.24 K 158.92 K
5	RCOOR´ R˝OH RCOOR˝ R´OH	3-methylbutyl acetate butanol butyl acetate 3-methylbutanol	-3.84 -1.74 148.83 K 154.98 K

4.4. Case Study 4

The aim of this case study is to compare the performance of four different tree based ensemble machine learning algorithms with respect to the prediction of rate constant of Diels-Alder reaction. The performance evaluation is conducted in terms of the R^2 and Q^2 values. Additionally, the performance is compared with the model developed by Datta et al. (2016a). A diverse data set pertaining to Diels-Alder reaction that consists of 38 different dienophiles, 19 dienes and 10 solvents was obtained from the work of Datta et al. (2016a). One sixth of the data set was utilized for testing the ensemble models and the remaining portion was utilized to train the models. The models were implemented, trained and tested using the R™ software. The chemical species involved in the Diels-Alder reaction were designed using the Avogadro™ software. The chemical

structures were optimized using MMFF94s, a built-in geometry optimization algorithm of Avogadro™. The optimized geometries were saved in the form of mol files. Next, these mol files were utilized as input to the Dragon™ 6 software to calculate the values of various descriptors belonging to the class of connectivity indices. In total, the values of 111 connectivity index based descriptors was calculated, 37 each for the diene, dienophile and the organic solvent. This initial set of 111 descriptors was reduced to a set of 30 descriptors using the RReliefF algorithm of Robnik-Šikonja and Kononenko (1997), for model development. RReliefF was implemented using the 'COREl earn' package in R. RReliefF falls in the category of filter methods of feature selection. In our work, for all the ensemble methods, we utilized 500 trees to predict the rate constant and we set the value of number of features evaluated at each node as 10. Additionally, we implemented random forests using the 'RandomForest' package in R. On the other hand, regularized random forests, extremely randomized trees and gradient boosted regression trees were implemented using the 'caret' package in R. The R^2 and Q^2 values obtained for the four tree based ensemble methods are listed in Table 4.12. As shown in Table 4.12, with respect to the training set, all the utilized ensemble methods except gradient boosted regression trees, performed at least comparably or better than the hybrid GA-DT of Datta et al. (2016a). However, the performance was lower, when compared to the hybrid GA-DT, on the test set. Gradient boosted regression trees, in general fared poorly. This could be due to the small sample size used in our work. Randomization-based methods performed well overall and seem as promising and scalable alternatives for prediction of rate constant of reaction. Generally, with regards to QSPR development, models with Q^2 value greater than 0.5 are considered to be acceptable. As pointed in section 3.4, decision tree based methods do not generate parametric models like in the case

of multiple linear regression. Decision tree based methods generate nonparametric models i.e. they do not make any assumptions about the distribution of the data.

Table 4.12: R^2 and Q^2 values of different ensemble methods

Ensemble Method	R^2	Q^2
Hybrid GA-DT	0.81	0.86
Random Forests	0.81	0.76
Regularized Random Forests	0.81	0.74
Extremely Randomized Trees	0.91	0.73
Gradient Boosted Trees	0.57	0.48

5. Conclusions and Future Directions

In this work, we have proposed revised structural feasibility constraints. The structural constraints fully take into account the overlapping of neighborhoods of bonding atoms. This avoids any mismatch that might occur at higher height in signatures that have the potential to bond. For a special case involving GCMs, an additional structural feasibility constraint has also been proposed. The constraint is applicable when occurrence number of all signatures are expressed in terms of the occurrence number of signatures of either the second highest height or the highest height. The highest height signature owes its presence to groups from GCMs only, in this case. In order for the designed reactants and products to comply with the reaction mechanism, a methodology has been proposed to relate the structures of reactants and products in accordance with the mechanism. In this methodology, we related the occurrence numbers of signatures of reactants and the occurrence numbers of signatures of the products. This relationship is captured in the RABS formulation. After relating the reactants and products structurally in terms of signatures, three design scenarios were addressed in this work. In the first scenario, we were concerned only with the optimization of dominant properties of products. This was extended to the optimization of properties dependent on structures of both reactants and products, in the second design scenario. A multi-objective optimization based approach was then formulated in the third design scenario to address the optimization of conflicting dominant properties of each reactant and product. Based on the insights developed for the design scenarios, we solved case studies, one for each scenario. The methodologies developed for each design scenario is not dependent on the number of reactants and products, and it also takes into account the reaction mechanism that may be involved.

Besides addressing different design scenarios, we also evaluated different tree based ensemble machine learning algorithms so that they could be used to predict properties relevant in reactive systems. Specifically, we were interested in predicting the rate constant of a reaction so that reactants and products can be designed to optimize the rate of a reaction. In our evaluation we found that, except gradient boosted regression trees, all methods were at least comparable or better than the hybrid GA-DT algorithm of Datta et al. (2016a) when compared on the basis of the training set. Extremely randomized trees with a R^2 value of 0.91 outperformed all other methods. On the basis of the test set, however, the hybrid GA-DT algorithm outperformed all methods on the training set with the highest value of Q^2 being 0.86. Overall, randomization-based methods' performance was comparable to the hybrid GA-DT method. All the calculated metrics in our work were based on the small concatenated data set of Datta et al. (2016a).

While the approaches showcased in this work are successful in addressing the limitations of the currently available methodologies for CAMD of reactants and products, there is further scope of extension in the approaches developed. For the future, the following extensions are proposed:

5.1. Design at Different Temperatures and Pressures

The three design scenarios considered in our work involve an assumption of the conditions being standard state conditions. However, many reactions in practice do not occur at standard conditions. Our work would benefit from the inclusion of operating temperatures and pressures as variables. This can be achieved by first generating models that are applicable over a large range of temperatures and pressures. Necessary data will have to be first gathered to generate such models. Once the models have been developed, the design methodologies developed so far will

have to be extended to design reactants and products within the temperature and pressure ranges of interest.

5.2. Modelling and Maximization of Rate Constant of Reactions

While the second design scenario addressed the optimization of thermodynamic properties of reactions, the maximization of rate of these reactions is also of prime importance. In the past, the CAMD of solvents has been carried out to maximize the rate constant of reactions. We would like to develop a CAMD methodology which takes into account the structural relationship between the rate constant and structures of reactants and molecular solvents. To partly address this design challenge, we developed QSPRs to predict the value of the rate constant for Diels-Alder reaction using the molecular descriptors of reactants and solvents as input. Models for other reactions being studied can be similarly developed. However, availability of data is a concern. In general, once the models have been developed, the design methodologies developed so far will have to be extended to design reactants, products and solvents such that the rate of the reaction is maximized. A further possible extension would be to also take into account the effect of temperature on the rate constant of the reaction. So far, the available CAMD methodologies and the models utilized to design solvents for reactions have not taken the variation of temperature into consideration. Standard state conditions have been assumed in these available methodologies.

5.3. Modelling and Design of Reactants, Products and Ionic Liquids

While ample research has been carried out to design molecular solvents that maximize the rate constant of a reaction, there has been no design study that formulates a procedure to design ionic liquids such that the rate of a reaction is maximized. Ionic liquids have recently generated enhanced interest because they have attractive properties such as low vapor pressure and high thermal stability. They are also known to influence the rate of reactions, just like molecular solvents. Designing an ionic liquid that influences the rate of reactions would involve the selection of the cation and the anion that makes up the ionic liquids. To carry out design of such solvents, first, property models will have to be generated that relate the structures of ionic liquids to that of the rate constant of the reaction. So far, to the best of our knowledge, such models have not been generated using molecular descriptors. A further extension would be to study the influence of structures of reactants, products and ionic liquids simultaneously on the rate constant of a reaction. Once, this is achieved, a CAMD methodology can be developed such that the design of reactants, products and ionic liquids can be carried out. Additionally, the effect of temperature can also be studied once the aforementioned research gaps have been filled.

6. References

2.5D Descriptors and Related Approaches. (2010). In *Three dimensional QSAR* (pp. 205–252). CRC Press. https://doi.org/doi:10.1201/b10419-10

Adjiman, C. S., Galindo, A., & Jackson, G. (2014). Molecules Matter: The Expanding Envelope of Process Design. Computer Aided Chemical Engineering, Volume 34, pp. 55–64. Elsevier. https://doi.org/10.1016/B978-0-444-63433-7.50007-9

Algamal, Z. Y., Lee, M. H., Al-Fakih, A. M., & Aziz, M. (2015). High-dimensional QSAR prediction of anticancer potency of imidazo[4,5-b]pyridine derivatives using adjusted adaptive LASSO. *Journal of Chemometrics*, *29*(10), 547–556. https://doi.org/10.1002/cem.2741

Baba, H., Takahara, J., Yamashita, F., & Hashida, M. (2015). Modeling and Prediction of Solvent Effect on Human Skin Permeability using Support Vector Regression and Random Forest. *Pharmaceutical Research*, *32*(11), 3604–3617. https://doi.org/10.1007/s11095-015-1720-4

Balakrishnan, R., & Ranganathan, K. (2012). Trees. In *A Textbook of Graph Theory SE - 4* (pp. 73–95). Springer New York. https://doi.org/10.1007/978-1-4614-4529-6_4

Baskin, I. (2008). Chapter 1 Fragment Descriptors in SAR/QSAR/QSPR Studies, Molecular Similarity Analysis and in Virtual Screening. In A. Varnek (Ed.), *Chemoinformatics Approaches to Virtual Screening* (pp. 1–43). The Royal Society of Chemistry. https://doi.org/10.1039/9781847558879-00001

Berk, R. A. (2016). Some Other Procedures Briefly BT - Statistical Learning from a Regression Perspective. In R. A. Berk (Ed.) (pp. 311–323). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-44048-4_8

Bolón-Canedo, V., Sánchez-Maroño, N., & Alonso-Betanzos, A. (2015). Foundations of Feature Selection - Feature Selection for High-Dimensional Data. In V. Bolón-Canedo, N. Sánchez-Maroño, & A. Alonso-Betanzos (Eds.) (pp. 13–28). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-21858-8_2

Bonami, P., Kilinç, M., & Linderoth, J. (2012). Algorithms and Software for Convex Mixed Integer Nonlinear Programs. In J. Lee & S. Leyffer (Eds.), *Mixed Integer Nonlinear Programming - 1* (Vol. 154, pp. 1–39). Springer New York. https://doi.org/10.1007/978-1-4614-1927-3_1

Bonilla-Petriciolet, A., & Rangaiah, G. P. (2013). Introduction. In *Multi-Objective Optimization in Chemical Engineering* (pp. 1–16). John Wiley & Sons Ltd. https://doi.org/10.1002/9781118341704.ch1

Breiman, L. (2001). Random Forests. *Machine Learning*, *45*(1), 5–32. https://doi.org/10.1023/A:1010933404324

Brignole, E. A., Bottini, S., & Gani, R. (1986). A strategy for the design and selection of solvents for separation processes. *Fluid Phase Equilibria*, *29*, 125–132. https://doi.org/10.1016/0378-3812(86)85016-6

Brown, A. C., & Fraser, T. R. (1868). On the Connection between Chemical Constitution and Physiological Action; with special reference to the Physiological Action of the Salts of the Ammonium Bases derived from Strychnia, Brucia, Thebaia, Codeia, Morphia, and Nicotia. *Journal of Anatomy and Physiology*, *2*(2), 224–242. Retrieved from http://www.ncbi.nlm.nih.gov/pmc/articles/PMC1318606/

Brown, W. M., Martin, S., Rintoul, M. D., & Faulon, J. L. (2006). Designing novel polymers with targeted properties using the signature molecular descriptor. *Journal of Chemical Information and Modeling*, *46*(2), 826–835. https://doi.org/10.1021/ci0504521

Calibration. (2009). In *Introduction to Multivariate Statistical Analysis in Chemometrics*. CRC Press. https://doi.org/doi:10.1201/9781420059496.ch4

Cao, D.-S., Xu, Q.-S., Liang, Y.-Z., Zhang, L.-X., & Li, H.-D. (2010). The boosting: A new idea of building models. *Chemometrics and Intelligent Laboratory Systems*, *100*(1), 1–11. https://doi.org/10.1016/j.chemolab.2009.09.002

Ceder, G. (2013). How Supercomputers Will Yield a Golden Age of Materials Science. Retrieved November 8, 2015, from http://www.scientificamerican.com/article/how-supercomputers-will-yield-a-golden-age-of-materials-science/

Chaudry, U. A., & Popelier, P. L. A. (2003). Ester Hydrolysis Rate Constant Prediction from Quantum Topological Molecular Similarity Descriptors. *The Journal of Physical Chemistry A*, *107*(22), 4578–4582. https://doi.org/10.1021/jp034272a

Chemmangattuvalappil, N. G., & Eden, M. R. (2013). A Novel Methodology for Property-Based Molecular Design Using Multiple Topological Indices. *Industrial & Engineering Chemistry Research*, *52*(22), 7090–7103. https://doi.org/10.1021/ie302516v

Chemmangattuvalappil, N. G., & Ng, D. K. S. (2013). A systematic methodology for optimal product design in an integrated biorefinery. *Computer Aided Chemical Engineering*, *32*, 91–96. https://doi.org/10.1016/B978-0-444-63234-0.50016-6

Chemmangattuvalappil, N. G., Roberts, C. B., & Eden, M. R. (2012). Signature Descriptors for Process and Molecular Design in Reactive Systems. *Computer Aided Chemical Engineering*, Volume 31, pp. 1356–1360). Elsevier. https://doi.org/10.1016/B978-0-444-59506-5.50102-4

Chemmangattuvalappil, N. G., Solvason, C. C., Bommareddy, S., & Eden, M. R. (2010). Reverse problem formulation approach to molecular design using property operators based on signature descriptors. *Computers and Chemical Engineering*, *34*(12), 2062–2071. https://doi.org/10.1016/j.compchemeng.2010.07.009

Chen, J. J. F., & Visco Jr., D. P. (2017). Developing an in silico pipeline for faster drug candidate discovery: Virtual high throughput screening with the Signature molecular descriptor using support vector machine models. *Chemical Engineering Science*, *159*, 31–42. https://doi.org/10.1016/j.ces.2016.02.037

Cheng, H. C., & Wang, F. S. (2007). Trade-off optimal design of a biocompatible solvent for an extractive fermentation process. *Chemical Engineering Science*, *62*(16), 4316–4324. https://doi.org/10.1016/j.ces.2007.05.010

Cheng, H. C., & Wang, F. S. (2010). Computer-aided biocompatible solvent design for an integrated extractive fermentation-separation process. *Chemical Engineering Journal*, *162*(2), 809–820. https://doi.org/10.1016/j.cej.2010.06.018

Churchwell, C. J., Rintoul, M. D., Martin, S., Visco, D. P., Kotu, A., Larson, R. S., … Faulon, J. L. (2004). The signature molecular descriptor: 3. Inverse-quantitative structure-activity relationship of ICAM-1 inhibitory peptides. *Journal of Molecular Graphics and Modelling*, *22*(4), 263–273. https://doi.org/10.1016/j.jmgm.2003.10.002

Cichosz, P. (2015). Regression trees. In *Data Mining Algorithms* (pp. 261–294). John Wiley & Sons, Ltd. https://doi.org/10.1002/9781118950951.ch9

Cignitti, S., Zhang, L., & Gani, R. (2015). Computer-aided Framework for Design of Pure, Mixed and Blended Products. *Computer Aided Chemical Engineering*, Volume 37, pp. 2093–2098). Elsevier. https://doi.org/10.1016/B978-0-444-63576-1.50043-1

Cismondi, M., & Brignole, E. A. (2004). Molecular Design of Solvents: An Efficient Search Algorithm for Branched Molecules. *Industrial & Engineering Chemistry Research*, *43*(3), 784–790. https://doi.org/10.1021/ie0340140

Cortes, C., & Vapnik, V. (1995). Support-Vector Networks. *Machine Learning*, *20*(3), 273–297. https://doi.org/10.1023/A:1022627411411

Cronin, M. T. D., Jaworska, J. S., Walker, J. D., Comber, M. H. I., Watts, C. D., & Worth, A. P. (2003). Use of QSARs in international decision-making frameworks to predict health effects of chemical substances. *Environmental Health Perspectives*, *111*(10), 1391–1401. Retrieved from http://www.ncbi.nlm.nih.gov/pmc/articles/PMC1241622/

Datta, S., Dev, V. A., & Eden, M. R. (2016). Hybrid genetic algorithm-decision tree approach for rate constant prediction using structures of reactants and solvent for Diels-Alder reaction.

Computers and Chemical Engineering. https://doi.org/10.1016/j.compchemeng.2017.02.022

Datta, S., Dev, V. A., & Eden, M. R. (2016). Relating Reaction Rate Constant to Structures of Reactants and Solvent Using a Hybrid GA-DT Approach. *Computer Aided Chemical Engineering*, Vol. 38, pp. 2049–2054). https://doi.org/10.1016/B978-0-444-63428-3.50346-5

Datta, S., Dev, V.A., & Eden, M. (2017). Developing QSPR for predicting DNA drug binding affinity of 9-Anilinoacridine derivatives using correlation-based adaptive LASSO algorithm. *Computer Aided Chemical Engineering*. In press.

Datta, S., Herring, R. H., & Eden, M. R. (2015). Data Mining and Regression Algorithms for the Development of a QSPR Model Relating Solvent Structure and Ibuprofen Crystal Morphology. *Computer Aided Chemical Engineering*, *37*, 1439–1444. https://doi.org/10.1016/B978-0-444-63577-8.50085-1

De Vleeschouwer, F., Chankisjijev, A., Geerlings, P., & De Proft, F. (2015). Designing Stable Radicals with Highly Electrophilic or Nucleophilic Character: Thiadiazinyl as a Case Study. *European Journal of Organic Chemistry*, *2015*(3), 506–513. https://doi.org/10.1002/ejoc.201403198

De Vleeschouwer, F., Chankisjijev, A., Yang, W., Geerlings, P., & De Proft, F. (2013). Pushing the boundaries of intrinsically stable radicals: Inverse design using the thiadiazinyl radical as a template. *Journal of Organic Chemistry*, *78*(7), 3151–3158. https://doi.org/10.1021/jo400101d

De Vleeschouwer, F., Yang, W., Beratan, D. N., Geerlings, P., & De Proft, F. (2012). Inverse design of molecules with optimal reactivity properties: acidity of 2-naphthol derivatives. *Physical Chemistry Chemical Physics*, 16002–16013. https://doi.org/10.1039/c2cp42623d

Deng, H., & Runger, G. (2013). Gene selection with guided regularized random forest. *Pattern Recognition*, *46*(12), 3483–3489. https://doi.org/10.1016/j.patcog.2013.05.018

Dev, V. A., Chemmangattuvalappil, N. G., & Eden, M. R. (2014). Reactant Structure Generation by Signature Descriptors and Real Coded Genetic Algorithm. *Computer Aided Chemical Engineering*, *34*, 150–162. https://doi.org/10.1016/B978-0-444-63433-7.50033-X

Dev, V. A., Chemmangattuvalappil, N. G., & Eden, M. R. (2014). Structure generation of candidate reactants using signature descriptors. *Computer Aided Chemical Engineering*, *33*, 151–156. https://doi.org/10.1016/B978-0-444-63456-6.50026-0

Dev, V. A., Chemmangattuvalappil, N. G., & Eden, M. R. (2015). Designing Reactants and Products with Properties Dependent on Both Structures. *Computer Aided Chemical*

Engineering, Volume 37, pp. 1445–1450. Elsevier. https://doi.org/10.1016/B978-0-444-63577-8.50086-3

Dev, V. A., Chemmangattuvalappil, N. G., & Eden, M. R. (2016). Multi-Objective Computer-Aided Molecular Design of Reactants and Products. *Computer Aided Chemical Engineering*, Volume 38, pp. 2055–2060. https://doi.org/10.1016/B978-0-444-63428-3.50347-7

Dev, V.A., Datta, S., Chemmangattuvalappil, N., & Eden, M. (2017). Comparison of Tree Based Ensemble Machine Learning Methods for Prediction of Rate Constant of Diels-Alder Reaction. *Computer Aided Chemical Engineering*. In press.

Diels, O., & Alder, K. (1928). Synthesen in der hydroaromatischen Reihe. *Justus Liebigs Ann Chem*, *460*. https://doi.org/10.1002/jlac.19284600106

Ding, S., Zhu, H., Jia, W., & Su, C. (2012). A survey on feature extraction for pattern recognition. *Artificial Intelligence Review*, *37*(3), 169–180. https://doi.org/10.1007/s10462-011-9225-y

Du, K.-L., & Swamy, M. N. S. (2014). Support Vector Machines BT - Neural Networks and Statistical Learning. In K.-L. Du & M. N. S. Swamy (Eds.) (pp. 469–524). London: Springer London. https://doi.org/10.1007/978-1-4471-5571-3_16

Ekins, S. (2016). The Next Era: Deep Learning in Pharmaceutical Research. *Pharmaceutical Research*, *33*(11), 2594–2603. https://doi.org/10.1007/s11095-016-2029-7

Faria, R. P. V, Pereira, C. S. M., Silva, V. M. T. M., Loureiro, J. M., & Rodrigues, A. E. (2013). Glycerol valorisation as biofuels: Selection of a suitable solvent for an innovative process for the synthesis of GEA. *Chemical Engineering Journal*, *233*, 159–167. https://doi.org/10.1016/j.cej.2013.08.035

Faulon, J. L., Churchwell, C. J., & Visco, D. P. (2003). The signature molecular descriptor. 2. Enumerating molecules from their extended valence sequences. *Journal of Chemical Information and Computer Sciences*, *43*(3), 721–734. https://doi.org/10.1021/ci020346o

Faulon, J. L., Visco, D. P., & Pophale, R. S. (2003). The signature molecular descriptor. 1. Using extended valence sequences in QSAR and QSPR studies. *Journal of Chemical Information and Computer Sciences*, *43*(3), 707–720. https://doi.org/10.1021/ci020345w

Faulon, J.-L., Faulon, J.-L., Collins, M. J., Collins, M. J., Carr, R. D., & Carr, R. D. (2004). The signature molecular descriptor. 4. Canonizing molecules using extended valence sequences. *Journal of Chemical Information and Computer Sciences*, *44*(2), 427–36. https://doi.org/10.1021/ci0341823

Feature Extraction and Feature Selection. (2011). In *Introduction to Pattern Recognition and Machine Learning* (Vol. Volume 5, pp. 75–110). Co-Published with Indian Institute of Science (IISc), Bangalore, India. https://doi.org/doi:10.1142/9789814335461_0003

Ferreira, L. S., & Trierweiler, J. O. (2009). Modeling and simulation of the polymeric nanocapsule formation process. *IFAC Proceedings Volumes (IFAC-PapersOnline)*, *7*(PART 1), 405–410. https://doi.org/10.1002/aic

Florios, K., & Mavrotas, G. (2014). Generation of the exact Pareto set in Multi-Objective Traveling Salesman and Set Covering Problems. *Applied Mathematics and Computation*, *237*, 1–19. https://doi.org/10.1016/j.amc.2014.03.110

Folić, M., Adjiman, C. S., & Pistikopoulos, E. N. (2008). Computer-aided solvent design for reactions: Maximizing product formation. *Industrial and Engineering Chemistry Research*, *47*(15), 5190–5202. https://doi.org/10.1021/ie0714549

Folić, M., Gani, R., Jiménez-González, C., & Constable, D. J. C. (2008). Systematic Selection of Green Solvents for Organic Reacting Systems* * Supported by PRISM FP6 Marie Curie Research Training Network (MRTN-CT-2004-512233). *Chinese Journal of Chemical Engineering*, *16*(3), 376–383. https://doi.org/10.1016/S1004-9541(08)60092-0

Friedman, J. H. (2001). Greedy Function Approximation: A Gradient Boosting Machine. *The Annals of Statistics*, *29*(5), 1189–1232. Retrieved from http://www.jstor.org/stable/2699986

Front Matter. (2011). In *Process Systems Engineering* (pp. I–XVII). Wiley-VCH Verlag GmbH & Co. KGaA. https://doi.org/10.1002/9783527631315.fmatter

Gani, R., & Brignole, E. A. (1983). Molecular design of solvents for liquid extraction based on UNIFAC. *Fluid Phase Equilibria*, *13*, 331–340. https://doi.org/10.1016/0378-3812(83)80104-6

Gani, R., & Ng, K. M. (2015). Product design – Molecules, devices, functional products, and formulated products. *Computers & Chemical Engineering*, *81*, 70–79. https://doi.org/10.1016/j.compchemeng.2015.04.013

Gani, R., Achenie, L. E. K., & Venkatasubramanian, V. (2002). Chapter 1: Introduction to CAMD. In R. G. and V. V. B. T.-C. A. C. E. Luke E.K. Achenie (Ed.), *Computer Aided Molecular Design: Theory and Practice* (Vol. Volume 12, pp. 3–21). Elsevier. https://doi.org/10.1016/S1570-7946(03)80003-2

Gani, R., Gómez, P. A., Folić, M., Jiménez-González, C., & Constable, D. J. C. (2008). Solvents in organic synthesis: Replacement and multi-step reaction systems. *Computers and Chemical Engineering*, *32*(10), 2420–2444. https://doi.org/10.1016/j.compchemeng.2008.01.006

Gani, R., Jiménez-González, C., & Constable, D. J. C. (2005). Method for selection of solvents for promotion of organic reactions. *Computers and Chemical Engineering*, *29*(7), 1661–1676. https://doi.org/10.1016/j.compchemeng.2005.02.021

Geladi, P., & Kowalski, B. R. (1986). Partial least-squares regression: a tutorial. *Analytica Chimica Acta*, *185*, 1–17. https://doi.org/10.1016/0003-2670(86)80028-9

Geurts, P., Ernst, D., & Wehenkel, L. (2006). Extremely randomized trees. *Machine Learning*, *63*(1), 3–42. https://doi.org/10.1007/s10994-006-6226-1

Glavič, P. (2012). Thirty Years of International Symposia on Process Systems Engineering. *Current Opinion in Chemical Engineering*, *1*(4), 421–429. https://doi.org/10.1016/j.coche.2012.10.002

Goodarzi, M., Dejaegher, B., & Heyden, Y. Vander. (n.d.). Feature Selection Methods in QSAR Studies. *Journal of AOAC International*. Retrieved from http://www.ingentaconnect.com/content/aoac/jaoac/2012/00000095/00000003/art00009

Goodarzi, M., Duchowicz, P. R., Wu, C. H., Fernandez, F. M., & Castro, E. A. (2009). New hybrid genetic based Support Vector Regression as QSAR approach for analyzing flavonoids-GABA(A) complexes. *J Chem Inf Model*, *49*(6), 1475–1485. Retrieved from http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Citation&list_uids=19492793

Grossmann, I. E., & Westerberg, a W. (2000). Research challenges in Process Systems Engineering. *AIChE Journal*, *46*(9), 1700–1703. https://doi.org/Cited By (since 1996) 103\nExport Date 6 March 2013

Guyon, I., & Elisseeff, A. (2006). Feature Extraction, Foundations and Applications: An introduction to feature extraction. *Studies in Fuzziness and Soft Computing*, *207*, 1–25. https://doi.org/10.1007/978-3-540-35488-8_1

Guyon, I., Elisseeff, A., & De, A. M. (2003). An Introduction to Variable and Feature Selection. *Journal of Machine Learning Research*, *3*, 1157–1182. https://doi.org/10.1016/j.aca.2011.07.027

Hall, L. H., Kier, L. B., & Murray, W. J. (1975). Molecular connectivity II: Relationship to water solubility and boiling point. *Journal of Pharmaceutical Sciences*, *64*(12), 1974–1977. https://doi.org/10.1002/jps.2600641215

Hansch, C., & Fujita, T. (1964). p-σ-π Analysis. A Method for the Correlation of Biological Activity and Chemical Structure. *Journal of the American Chemical Society*, *86*(8), 1616–1626. https://doi.org/10.1021/ja01062a035

HANSCH, C., MALONEY, P. P., FUJITA, T., & MUIR, R. M. (1962). Correlation of Biological Activity of Phenoxyacetic Acids with Hammett Substituent Constants and Partition Coefficients. *Nature*, *194*(4824), 178–180. Retrieved from http://dx.doi.org/10.1038/194178b0

Härdle, W. K., & Simar, L. (2015). Variable Selection BT - Applied Multivariate Statistical Analysis. In W. K. Härdle & L. Simar (Eds.) (pp. 281–304). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-662-45171-7_9

Harper, P. M., & Gani, R. (2000). A multi-step and multi-level approach for computer aided molecular design. *Computers & Chemical Engineering*, *24*(2–7), 677–683. https://doi.org/10.1016/S0098-1354(00)00410-5

Harper, P. M., Hostrup, H., & Gani, R. (2002). Chapter 6: A hybrid CAMD method. *Computer Aided Molecular Design: Theory and Practice* (Volume 12, pp. 129–165). Elsevier. https://doi.org/10.1016/S1570-7946(03)80008-1

Hechinger, M., Leonhard, K., & Marquardt, W. (2012). What is wrong with quantitative structure-property relations models based on three-dimensional descriptors? *Journal of Chemical Information and Modeling*, *52*(8), 1984–1993. https://doi.org/10.1021/ci300246m

Herring, R. H., & Eden, M. R. (2015). Evolutionary algorithm for de novo molecular design with multi-dimensional constraints. *Computers & Chemical Engineering*. https://doi.org/10.1016/j.compchemeng.2015.06.012

Hill, M. (2009). Chemical Product Engineering—The third paradigm. *Computers & Chemical Engineering*, *33*(5), 947–953. https://doi.org/10.1016/j.compchemeng.2008.11.013

Hong, H., Slavov, S., Ge, W., Qian, F., Su, Z., Fang, H., … Tong, W. (2012). Mold2 Molecular Descriptors for QSAR. In *Statistical Modelling of Molecular Descriptors in QSAR/QSPR* (pp. 65–109). Wiley-VCH Verlag GmbH & Co. KGaA. https://doi.org/10.1002/9783527645121.ch3

Hukkerikar, A. S., Sarup, B., Ten Kate, A., Abildskov, J., Sin, G., & Gani, R. (2012). Group-contribution + (GC +) based estimation of properties of pure components: Improved property estimation and uncertainty analysis. *Fluid Phase Equilibria*, *321*, 25–43. https://doi.org/10.1016/j.fluid.2012.02.010

Index. (2011). In *Process Systems Engineering* (pp. 307–317). Wiley-VCH Verlag GmbH & Co. KGaA. https://doi.org/10.1002/9783527631315.index

Itskowitz, P., & Tropsha, A. (2005). k Nearest Neighbors QSAR Modeling as a Variational Problem: Theory and Applications. *Journal of Chemical Information and Modeling*, *45*(3), 777–785. https://doi.org/10.1021/ci049628+

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An introduction to statistical learning: with applications in R*. *Springer texts in statistics* (Vol. XIV). https://doi.org/10.1007/978-1-4614-7138-7

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). Linear Model Selection and Regularization BT - An Introduction to Statistical Learning: with Applications in R. In G. James, D. Witten, T. Hastie, & R. Tibshirani (Eds.) (pp. 203–264). New York, NY: Springer New York. https://doi.org/10.1007/978-1-4614-7138-7_6

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). Tree-Based Methods - An Introduction to Statistical Learning: with Applications in R. In G. James, D. Witten, T. Hastie, & R. Tibshirani (Eds.) (pp. 303–335). New York, NY: Springer New York. https://doi.org/10.1007/978-1-4614-7138-7_8

Jankowski, M. D., Henry, C. S., Broadbelt, L. J., & Hatzimanikatis, V. (2008). Group contribution method for thermodynamic analysis of complex metabolic networks. *Biophysical Journal*, *95*(3), 1487–1499. https://doi.org/10.1529/biophysj.107.124784

Jayaseelan, K. V., Moreno, P., Truszkowski, A., Ertl, P., & Steinbeck, C. (2012). Natural product-likeness score revisited: an open-source, open-data implementation. *BMC Bioinformatics*, *13*(1), 106. https://doi.org/10.1186/1471-2105-13-106

Jinich, A., Rappoport, D., Dunn, I., Sanchez-lengeling, B., Olivares-amaya, R., Noor, E., & Even, A. B. (2014). Quantum Chemical Approach to Estimating the Thermodynamics of Metabolic Reactions, 1–6. https://doi.org/10.1038/srep07022

Karthikeyan, M., & Vyas, R. (2014). Machine Learning Methods in Chemoinformatics for Drug Discovery BT - Practical Chemoinformatics. In M. Karthikeyan & R. Vyas (Eds.) (pp. 133–194). New Delhi: Springer India. https://doi.org/10.1007/978-81-322-1780-0_3

Kayello, H. M., Tadisina, N. K. R., Shlonimskaya, N., Biernacki, J. J., & Visco, D. P. (2014). An application of computer-aided molecular design (CAMD) using the signature molecular descriptor - Part 1. Identification of surface tension reducing agents and the search for shrinkage reducing admixtures. *Journal of the American Ceramic Society*, *97*(2), 365–377. https://doi.org/10.1111/jace.12453

Kendon, V., Sebald, A., & Stepney, S. (2015). Heterotic computing: exploiting hybrid computational devices. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, *373*(2046). Retrieved from http://rsta.royalsocietypublishing.org/content/373/2046/20150091.abstract

Kew, W., & Mitchell, J. B. O. (2015). Greedy and Linear Ensembles of Machine Learning Methods Outperform Single Approaches for QSPR Regression Problems. *Molecular Informatics*, *34*(9), 634–647. https://doi.org/10.1002/minf.201400122

Kier, L. B., Hall, L. H., Murray, W. J., & Randi, M. (1975). Molecular connectivity I: Relationship to nonspecific local anesthesia. *Journal of Pharmaceutical Sciences*, *64*(12), 1971–1974. https://doi.org/10.1002/jps.2600641214

Kier, L. B., Murray, W. J., & Hall, L. H. (1975). Molecular connectivity. 4. Relations to biological activities. *Journal of Medicinal Chemistry*, *18*(12), 1272–1274. https://doi.org/10.1021/jm00246a025

Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by Simulated Annealing. *Science*, *220*(4598), 671 LP-680. Retrieved from http://science.sciencemag.org/content/220/4598/671.abstract

Klamt, A. (2011). The COSMO and COSMO-RS solvation models. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, *1*(5), 699–709. https://doi.org/10.1002/wcms.56

Klatt, K.-U., & Marquardt, W. (2009). Perspectives for process systems engineering—Personal views from academia and industry. *Computers & Chemical Engineering*, *33*(3), 536–550. https://doi.org/10.1016/j.compchemeng.2008.09.002

Kohavi, R., & John, G. H. (1997). Wrappers for feature subset selection. *Artificial Intelligence*, *97*(1–2), 273–324. https://doi.org/10.1016/S0004-3702(97)00043-X

Kononenko, I., & Kukar, M. (2007). Chapter 7 - Data Preprocessing - Machine Learning and Data Mining (pp. 181–211). Woodhead Publishing. https://doi.org/https://doi.org/10.1533/9780857099440.181

Kononenko, I., Šimec, E., & Robnik-Šikonja, M. (1997). Overcoming the myopia of inductive learning algorithms with RELIEFF. *Applied Intelligence*, *7*(1), 39–55. https://doi.org/10.1023/A:1008280620621

Kuhn, M. (2016). Quantitative-Structure Activity Relationship Modeling and Cheminformatics - Nonclinical Statistics for Pharmaceutical and Biotechnology Industries. In L. Zhang (Ed.) (pp. 141–155). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-23558-5_6

Kuhn, M., & Johnson, K. (2013). Regression Trees and Rule-Based Models - Applied Predictive Modeling. In M. Kuhn & K. Johnson (Eds.) (pp. 173–220). New York, NY: Springer New York. https://doi.org/10.1007/978-1-4614-6849-3_8

Leardi, R. (2001). Genetic algorithms in chemometrics and chemistry: a review. *Journal of Chemometrics*, *15*(7), 559–569. https://doi.org/10.1002/cem.651

Li, H., Liang, Y., & Xu, Q. (2009). Support vector machines and its applications in chemistry. *Chemometrics and Intelligent Laboratory Systems*, *95*(2), 188–198. https://doi.org/http://dx.doi.org/10.1016/j.chemolab.2008.10.007

Liew, C. Y., Ma, X. H., Liu, X., & Yap, C. W. (2009). SVM Model for Virtual Screening of Lck Inhibitors. *Journal of Chemical Information and Modeling*, *49*(4), 877–885. https://doi.org/10.1021/ci800387z

Lin, B., Chavali, S., Camarda, K., & Miller, D. C. (2005). Computer-aided molecular design using Tabu search. *Computers & Chemical Engineering*, *29*(2), 337–347. https://doi.org/10.1016/j.compchemeng.2004.10.008

Linke, P., & Kokossis, A. (2002). Simultaneous Synthesis and Design of Novel Chemicals and Chemical Process Flowsheets. *Computer Aided Chemical Engineering*, Volume 10, pp. 115–120). Elsevier. https://doi.org/10.1016/S1570-7946(02)80047-5

Luts, J., Ojeda, F., Van de Plas, R., De Moor, B., Van Huffel, S., & Suykens, J. A. K. (2010). A tutorial on support vector machine-based methods for classification problems in chemometrics. *Analytica Chimica Acta*, *665*(2), 129–145. https://doi.org/10.1016/j.aca.2010.03.030

MacGregor, J. F., & Kourti, T. (1995). Statistical process control of multivariate processes. *Control Engineering Practice*, *3*(3), 403–414. https://doi.org/10.1016/0967-0661(95)00014-L

Maggiora, G. M. (2014). Introduction to Molecular Similarity and Chemical Space. In K. Martinez-Mayorga & J. L. Medina-Franco (Eds.), *Foodinformatics SE - 1* (pp. 1–81). Springer International Publishing. https://doi.org/10.1007/978-3-319-10226-9_1

Marrero, J., & Gani, R. (2001). Group-contribution based estimation of pure component properties. *Fluid Phase Equilibria*, *183–184*, 183–208. https://doi.org/10.1016/S0378-3812(01)00431-9

Martin, T. M., & Young, D. M. (2001). Prediction of the acute toxicity (96-h LC50) of organic compounds to the fathead minnow (Pimephales promelas) using a group contribution method. *Chemical Research in Toxicology*, *14*(10), 1378–85. https://doi.org/10.1021/tx0155045

Mavrotas, G. (2009). Effective implementation of the ??-constraint method in Multi-Objective Mathematical Programming problems. *Applied Mathematics and Computation*, *213*(2), 455–465. https://doi.org/10.1016/j.amc.2009.03.037

Mavrotas, G., & Florios, K. (2013). An improved version of the augmented s-constraint method (AUGMECON2) for finding the exact pareto set in multi-objective integer programming

problems. *Applied Mathematics and Computation*, *219*(18), 9652–9669. https://doi.org/10.1016/j.amc.2013.03.002

Meir, R., & Rätsch, G. (2003). An Introduction to Boosting and Leveraging BT - Advanced Lectures on Machine Learning: Machine Learning Summer School 2002 Canberra, Australia, February 11–22, 2002 Revised Lectures. In S. Mendelson & A. J. Smola (Eds.) (pp. 118–183). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/3-540-36434-X_4

Miettinen, K. (1998). Theoretical Background. In *Nonlinear Multiobjective Optimization SE - 2* (Vol. 12, pp. 37–57). Springer US. https://doi.org/10.1007/978-1-4615-5563-6_2

Miettinen, K. (2008). Introduction to Multiobjective Optimization: Noninteractive Approaches. In J. Branke, K. Deb, K. Miettinen, & R. Słowiński (Eds.), *Multiobjective Optimization - 1* (Vol. 5252, pp. 1–26). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-88908-3_1

Miettinen, K., Ruiz, F., & Wierzbicki, A. (2008). Introduction to Multiobjective Optimization: Interactive Approaches. In J. Branke, K. Deb, K. Miettinen, & R. Słowiński (Eds.), *Multiobjective Optimization - 2* (Vol. 5252, pp. 27–57). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-88908-3_2

Mitchell, J. B. O. (2014). Machine learning methods in chemoinformatics. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, *4*(5), 468–481. https://doi.org/10.1002/wcms.1183

Mlinar, V. (2015). Utilization of inverse approach in the design of materials over nano- to macro-scale. *Annalen Der Physik*, *527*(3–4), 187–204. https://doi.org/10.1002/andp.201400190

Mu, L., & He, H. (2011). QSPR study of standard absolute entropies for gaseous organic compounds using novel molecular connectivity indexes and Ring parameter. *Thermochimica Acta*, *526*(1), 99–106. https://doi.org/10.1016/j.tca.2011.08.024

Nandi, S., Monesi, A., Drgan, V., Merzel, F., & Novič, M. (2013). Quantitative structure-activation barrier relationship modeling for Diels-Alder ligations utilizing quantum chemical structural descriptors. *Chemistry Central Journal*, *7*(1), 171. https://doi.org/10.1186/1752-153X-7-171

Natekin, A., & Knoll, A. (2013). Gradient boosting machines, a tutorial. *Frontiers in Neurorobotics*. Retrieved from http://journal.frontiersin.org/article/10.3389/fnbot.2013.00021

Ng, L. Y., Andiappan, V., Chemmangattuvalappil, N. G., & Ng, D. K. S. (2015). A systematic methodology for optimal mixture design in an integrated biorefinery. *Computers & Chemical Engineering*, *81*, 288–309. https://doi.org/10.1016/j.compchemeng.2015.04.032

Ng, L. Y., Chemmangattuvalappil, N. G., Dev, V. A., & Eden, M. R. (2017). *Chapter 1 – Mathematical Principles of Chemical Product Design and Strategies*. *Computer Aided Chemical Engineering* (Vol. 39). https://doi.org/10.1016/B978-0-444-63683-6.00001-0

Ng, L. Y., Chong, F. K., & Chemmangattuvalappil, N. G. (2015). Challenges and opportunities in computer-aided molecular design. *Computers & Chemical Engineering*, *81*, 115–129. https://doi.org/10.1016/j.compchemeng.2015.03.009

Niemi, J. B., & Niemi, G. J. (2015). Chapter 6 - Linear Regression, Model Averaging, and Bayesian Techniques for Predicting Chemical Activities from Structure A2 - Basak, Subhash C. In G. Restrepo & J. L. B. T.-A. in M. C. and A. Villaveces (Eds.) (pp. 125–147). Bentham Science Publishers. https://doi.org/10.1016/B978-1-68108-053-6.50006-5

Papadopoulos, a. I., & Linke, P. (2005). A Unified Framework for Integrated Process and Molecular Design. *Chemical Engineering Research and Design*, *83*(6), 674–678. https://doi.org/10.1205/cherd.04349

Papadopoulos, A. I., & Linke, P. (2006). Multiobjective molecular design for integrated process-solvent systems synthesis. *AIChE Journal*, *52*(3), 1057–1069. https://doi.org/10.1002/aic.10715

Papadopoulos, A. I., & Linke, P. (2009). Integrated solvent and process selection for separation and reactive separation systems. *Chemical Engineering and Processing: Process Intensification*, *48*(5), 1047–1060. https://doi.org/10.1016/j.cep.2009.02.004

Pompe, M., & Novič, M. (1999). Prediction of gas-chromatographic retention indices using topological descriptors. *J Chem Inf Comput Sci*, *39*. https://doi.org/10.1021/ci980036z

Potočnik, P., Grabec, I., Šetinc, M., & Levec, J. (2003). Hybrid modeling of kinetics for methanol synthesis - Soft Computing Approaches in Chemistry. In H. M. Cartwright & L. M. Sztandera (Eds.) (pp. 297–315). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-36213-5_11

Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning*, *1*(1), 81–106. https://doi.org/10.1007/BF00116251

Randic, M. (1975). Characterization of molecular branching. *Journal of the American Chemical Society*, *97*(23), 6609–6615. https://doi.org/10.1021/ja00856a001

Randic, M. (1984). On molecular identification numbers. *J Chem Inf Comput Sci*, *24*. https://doi.org/10.1021/ci00043a009

Reymond, J.-L. (2015). The Chemical Space Project. *Accounts of Chemical Research*, *48*(3), 722–730. https://doi.org/10.1021/ar500432k

Robnik-Šikonja, M., & Kononenko, I. (1997). An adaptation of Relief for attribute estimation in regression. *Machine Learning: Proceedings of the Fourteenth International Conference (ICML'97)*, *5*, 296–304. https://doi.org/10.1119/1.880454

Rokach, L. (2016). Decision forest: Twenty years of research. *Information Fusion*, *27*, 111–125. https://doi.org/10.1016/j.inffus.2015.06.005

Roy, K., Kar, S., & Das, R. (2015). QSAR/QSPR Modeling: Introduction. In *A Primer on QSAR/QSPR Modeling - 1* (pp. 1–36). Springer International Publishing. https://doi.org/10.1007/978-3-319-17281-1_1

Roy, K., Kar, S., & Das, R. N. (2015). Chapter 1 - Background of QSAR and Historical Developments - Understanding the Basics of QSAR for Applications in Pharmaceutical Sciences and Risk Assessment (pp. 1–46). Boston: Academic Press. https://doi.org/https://doi.org/10.1016/B978-0-12-801505-6.00001-6

Roy, K., Kar, S., & Das, R. N. (2015). Chapter 2 - Chemical Information and Descriptors - Understanding the Basics of QSAR for Applications in Pharmaceutical Sciences and Risk Assessment (pp. 47–80). Boston: Academic Press. https://doi.org/https://doi.org/10.1016/B978-0-12-801505-6.00002-8

Roy, K., Kar, S., & Das, R. N. (2015). Chapter 2 - Chemical Information and Descriptors. In K. R. K. N. B. T.-U. the B. of Q. for A. in P. S. and R. A. Das (Ed.) (pp. 47–80). Boston: Academic Press. https://doi.org/10.1016/B978-0-12-801505-6.00002-8

Roy, K., Kar, S., & Das, R. N. (2015). Chapter 4 - Topological QSAR. In K. R. K. N. B. T.-U. the B. of Q. for A. in P. S. and R. A. Das (Ed.) (pp. 103–149). Boston: Academic Press. https://doi.org/10.1016/B978-0-12-801505-6.00004-1

Roy, K., Kar, S., & Das, R. N. (2015). Chapter 6 - Selected Statistical Methods in QSAR BT - Understanding the Basics of QSAR for Applications in Pharmaceutical Sciences and Risk Assessment (pp. 191–229). Boston: Academic Press. https://doi.org/10.1016/B978-0-12-801505-6.00006-5

Sabljic, A., Guesten, H., Schoenherr, J., & Riederer, M. (1990). Modeling plant uptake of airborne organic chemicals. 1. Plant cuticle/water partitioning and molecular connectivity. *Environmental Science & Technology*, *24*(9), 1321–1326. https://doi.org/10.1021/es00079a004

Sargent, R. (2005). Process systems engineering: A retrospective view with questions for the future. *Computers & Chemical Engineering*, *29*(6), 1237–1241. https://doi.org/10.1016/j.compchemeng.2005.02.008

Schapire, R. E. (2003). The Boosting Approach to Machine Learning: An Overview - Nonlinear Estimation and Classification. In D. D. Denison, M. H. Hansen, C. C. Holmes, B. Mallick, & B. Yu (Eds.) (pp. 149–171). New York, NY: Springer New York. https://doi.org/10.1007/978-0-387-21579-2_9

Segall, M. (2012). Can we really do computer-aided drug design? *Journal of Computer-Aided Molecular Design*, *26*(1), 121–124. https://doi.org/10.1007/s10822-011-9512-3

Sheldon, T. J., Folić, M., & Adjiman, C. S. (2006). Solvent Design Using a Quantum Mechanical Continuum Solvation Model. *Industrial & Engineering Chemistry Research*, *45*(3), 1128–1140. https://doi.org/10.1021/ie050416r

Shivajirao, A., Sarup, B., Ten, A., Abildskov, J., & Engineering, B. (n.d.). Improved property estimation and uncertainty analysis Computer Aided Process-Product Engineering Center (CAPEC), Department of Chemical Vegetable Oil Technology Business Unit , Alfa Laval Copenhagen A / S , Maskinvej 5 , DK- Akzonobel Research , Develop.

Shlonimskaya, N., Biernacki, J. J., Kayello, H. M., & Visco, D. P. (2014). An application of computer-aided molecular design (CAMD) using the signature molecular descriptor-part 2. Evaluating newly identified surface tension-reducing substances for potential use as shrinkage-reducing admixtures. *Journal of the American Ceramic Society*, *97*(2), 378–385. https://doi.org/10.1111/jace.12677

Siedlecki, W., & Sklansky, J. (1989). A note on genetic algorithms for large-scale feature selection. *Pattern Recognition Letters*, *10*(5), 335–347. https://doi.org/10.1016/0167-8655(89)90037-8

Siougkrou, E., Galindo, A., & Adjiman, C. S. (2014). On the optimal design of gas-expanded liquids based on process performance. *Chemical Engineering Science*, *115*, 19–30. https://doi.org/10.1016/j.ces.2013.12.025

Smola, A. J., & Schölkopf, B. (2004). A tutorial on support vector regression. *Statistics and Computing*, *14*(3), 199–222. https://doi.org/10.1023/B:STCO.0000035301.49549.88

Šoškić, M., & Plavšić, D. (2005). Modeling the octanol-water partition coefficients by an optimized molecular connectivity index. *Journal of Chemical Information and Modeling*, *45*(4), 930–938. https://doi.org/10.1021/ci050024v

Stanescu, I., & Achenie, L. E. K. (2006). A theoretical study of solvent effects on Kolbe–Schmitt reaction kinetics. *Chemical Engineering Science*, *61*(18), 6199–6212. https://doi.org/10.1016/j.ces.2006.05.025

Stephanopoulos, G., & Reklaitis, G. V. (2011). Process systems engineering: From Solvay to modern bio- and nanotechnology.: A history of development, successes and prospects for the future. *Chemical Engineering Science*, *66*(19), 4272–4306. https://doi.org/10.1016/j.ces.2011.05.049

Strübing, H., Konstantinidis, S., Karamertzanis, P. G., Pistikopoulos, E. N., Galindo, A., & Adjiman, C. S. (2011). Computer-Aided Methodologies for the Design of Reaction Solvents. In *Process Systems Engineering* (pp. 267–305). Wiley-VCH Verlag GmbH & Co. KGaA. https://doi.org/10.1002/9783527631315.ch9

Struebing, H., Ganase, Z., Karamertzanis, P. G., Siougkrou, E., Haycock, P., Piccione, P. M., … Adjiman, C. S. (2013). Computer-aided molecular design of solvents for accelerated reaction kinetics. *Nature Chemistry*, *5*(11), 952–7. https://doi.org/10.1038/nchem.1755

Sumathi, R., Carstensen, H.-H., & Green, W. H. (2002). Reaction Rate Predictions Via Group Additivity. Part 3: Effect of Substituents with CH2 as the Mediator. *The Journal of Physical Chemistry A*, *106*(22), 5474–5489. https://doi.org/10.1021/jp013957c

Sutter, J. M., & Jurs, P. C. (1995). Chapter 5 Selection of molecular descriptors for quantitative structure-activity relationships. *Data Handling in Science and Technology*, *15*, 111–132. https://doi.org/10.1016/S0922-3487(06)80006-7

Svetnik, V., Wang, T., Tong, C., Liaw, A., Sheridan, R. P., & Song, Q. (2005). Boosting: An Ensemble Learning Tool for Compound Classification and QSAR Modeling. *Journal of Chemical Information and Modeling*, *45*(3), 786–799. https://doi.org/10.1021/ci0500379

Świderek, K., Tuñón, I., Moliner, V., & Bertran, J. (2015). Computational strategies for the design of new enzymatic functions. *Archives of Biochemistry and Biophysics*, *582*, 68–79. https://doi.org/10.1016/j.abb.2015.03.013

Tang, S.-. Y., Shi, J., & Guo, Q.-. X. (2012). Accurate prediction of rate constants of Diels–Alder reactions and application to design of Diels–Alder ligation. *Org Biomol Chem*, *10*. https://doi.org/10.1039/c2ob07079k

Tibshirani, R. (1996). Regression Shrinkage and Selection via the Lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, *58*(1), 267–288. Retrieved from http://www.jstor.org/stable/2346178

Todeschini, R., & Consonni, V. (2000). Frontmatter. In *Handbook of Molecular Descriptors* (pp. i–xxi). Wiley-VCH Verlag GmbH. https://doi.org/10.1002/9783527613106.fmatter

Todeschini, R., & Consonni, V. (2009). *Molecular Descriptors for Chemoinformatics, Revised and Enlarged Edition*. Weinheim: Wiley-VCH. https://doi.org/10.1002/9783527628766

van Speybroeck, V., Gani, R., & Meier, R. J. (2010). The calculation of thermodynamic properties of molecules. *Chemical Society Reviews*, *39*(5), 1764–1779. https://doi.org/10.1039/b809850f

Venkatraman, V., Dalby, A. R., & Yang, Z. R. (2004). Evaluation of Mutual Information and Genetic Programming for Feature Selection in QSAR. *Journal of Chemical Information and Computer Sciences*, *44*(5), 1686–1692. https://doi.org/10.1021/ci049933v

Visco, D. P., Pophale, R. S., Rintoul, M. D., & Faulon, J. L. (2002). Developing a methodology for an inverse quantitative structure-activity relationship using the signature molecular descriptor. *Journal of Molecular Graphics and Modelling*, *20*(6), 429–438. https://doi.org/10.1016/S1093-3263(01)00144-9

Visco, D.P. (2010). Computer-Aided Molecular Design. In *Handbook of Chemoinformatics Algorithms* (pp. 269–293). Chapman and Hall/CRC. https://doi.org/doi:10.1201/9781420082999-c9

Wang, M., Hu, X., Beratan, D. N., & Yang, W. (2006). Designing molecules by optimizing potentials. *Journal of the American Chemical Society*, *128*(10), 3228–3232. https://doi.org/10.1021/ja0572046

Wang, Y., & Achenie, L. E. K. (2002). A hybrid global optimization approach for solvent design. *Computers and Chemical Engineering*, *26*(10), 1415–1425. https://doi.org/10.1016/S0098-1354(02)00118-7

Wang, Y., & Achenie, L. E. K. (2002). Computer aided solvent design for extractive fermentation. *Fluid Phase Equilibria*, *201*(1), 1–18. https://doi.org/10.1016/S0378-3812(02)00073-0

Weis, D. C., & Visco, D. P. (2010). Computer-aided molecular design using the Signature molecular descriptor: Application to solvent selection. *Computers and Chemical Engineering*, *34*(7), 1018–1029. https://doi.org/10.1016/j.compchemeng.2009.10.017

Weis, D. C., Faulon, J. L., LeBorne, R. C., & Visco, D. P. (2005). The signature molecular descriptor. 5. The design of hydrofluoroether foam blowing agents using inverse-QSAR. *Industrial and Engineering Chemistry Research*, *44*(23), 8883–8891. https://doi.org/10.1021/ie050330y

Wentzell, P. D., & Vega Montoto, L. (2003). Comparison of principal components regression and partial least squares regression through generic simulations of complex mixtures. *Chemometrics and Intelligent Laboratory Systems*, *65*(2), 257–279. https://doi.org/10.1016/S0169-7439(02)00138-7

Weymuth, T., & Reiher, M. (2014). Gradient-driven molecule construction: An inverse approach applied to the design of small-molecule fixating catalysts. *International Journal of Quantum Chemistry*, *114*(13), 838–850. https://doi.org/10.1002/qua.24686

Weymuth, T., & Reiher, M. (2014). Inverse quantum chemistry: Concepts and strategies for rational compound design. *International Journal of Quantum Chemistry*, *114*(13), 823–837. https://doi.org/10.1002/qua.24687

Whitley, D. (1994). A genetic algorithm tutorial. *Statistics and Computing*, *4*(2), 65–85. https://doi.org/10.1007/BF00175354

Wicaksono, D. S., Mhamdi, A., & Marquardt, W. (2014). Computer-aided screening of solvents for optimal reaction rates. *Chemical Engineering Science*, *115*, 167–176. https://doi.org/10.1016/j.ces.2013.12.006

Wold, S., Kettaneh, N., & Tjessem, K. (1996). Hierarchical multiblock PLS and PC models for easier model interpretation and as an alternative to variable selection. *Journal of Chemometrics*, *10*(5–6), 463–482. https://doi.org/10.1002/(SICI)1099-128X(199609)10:5/6<463::AID-CEM445>3.0.CO;2-L

Xiao, D., Warnke, I., Bedford, J., & Batista, V. S. (2014). Chapter 1 Inverse molecular design for materials discovery. In *Chemical Modelling: Volume 10* (Vol. 10, pp. 1–31). The Royal Society of Chemistry. https://doi.org/10.1039/9781849737241-00001

Yee, L. C., & Wei, Y. C. (2012). Current Modeling Methods Used in QSAR/QSPR. In *Statistical Modelling of Molecular Descriptors in QSAR/QSPR* (pp. 1–31). Wiley-VCH Verlag GmbH & Co. KGaA. https://doi.org/10.1002/9783527645121.ch1

Zhang, J., Liu, Z., & Liu, W. (2014). QSPR study for prediction of boiling points of 2475 organic compounds using stochastic gradient boosting. *Journal of Chemometrics*, *28*(3), 161–167. https://doi.org/10.1002/cem.2587

Zhou, T., Lyu, Z., Qi, Z., & Sundmacher, K. (2015). Robust design of optimal solvents for chemical reactions—A combined experimental and computational strategy. *Chemical Engineering Science*, *137*, 613–625. https://doi.org/10.1016/j.ces.2015.07.010

Zhou, T., McBride, K., Zhang, X., Qi, Z., & Sundmacher, K. (2015). Integrated solvent and process design exemplified for a Diels–Alder reaction. *AIChE Journal*, *61*(1), 147–158. https://doi.org/10.1002/aic.14630

Zhou, T., Qi, Z., & Sundmacher, K. (2014). Model-based method for the screening of solvents for chemical reactions. *Chemical Engineering Science*, *115*, 177–185. https://doi.org/10.1016/j.ces.2013.11.020

Zou, H., & Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *67*(2), 301–320. https://doi.org/10.1111/j.1467-9868.2005.00503.x