Comparisons of Molecular Structure Generation Methods Based on Fragment Assemblies and Genetic Graphs

Philippe Gantzer, Benoit Creton, Carlos Nieto-Draghi

Comparisons of Molecular Structures Generation Methods Based on Fragments Assemblies and Genetic Graphs.

Philippe Gantzer, Benoit Creton, Carlos Nieto-Draghi*

IFP Energies nouvelles, 1 et 4 avenue de Bois-Préau, 92852 Rueil-Malmaison, France.

Abstract: The use of Quantitative Structure-Property Relationships (QSPR) helps in predicting molecular properties since now several decades, whilst the automatic design of new molecular structures is still emerging. The choice of algorithms to generate molecules is not obvious and is related to several factors such as desired chemical diversity (according to an initial dataset's content) and level of construction (the use of atoms, fragments, patterns-based methods). In this paper, we address the problem of molecular structure generation by revisiting two approaches: Fragments-based Methods (FM) and Genetic-based Methods (GM). We define a set of indices to compare generation methods on a specific task. New indices inform about the explored data space (coverage), compare how the data space is explored (representativeness) and quantifies the ratio of molecules satisfying requirements (generation specificity), without the use of a database composed of real chemicals as a reference. Those indices were employed to compare generations of molecules fulfilling a desired property criterion, evaluated by QSPR.

Introduction

R&D works in chemistry continuously contribute to the discovery of pathways for the production of new molecules. These molecules are then characterized by means of experimental analysis and calculations; new data are daily generated, supplementing chemistry databases. Databases store that information, *i.e.* the chemicals' names, structures, characteristics, properties... Not fewer than 60 databases are publicly available according to Apodaca[1]. Databases vary according to the molecules and the information they gather.[2] Considering these latter elements, identifying a compound satisfying a set of specific constraints becomes difficult due to the choice of databases, to their size, to the different kinds of data available and because of databases' different ways to store the information.

The field of Chemoinformatics focuses on processing and using the chemical information in a smart way to resolve chemical problems. Assuming that molecules with similar structures possess similar properties, it is possible to statistically correlate structures with properties; the so-obtained relations are called Quantitative Structure-Property Relationships (QSPR) or Quantitative Structure-Activity Relationships (QSAR)[3]. Hereafter, the acronym QSPR will also gather QSAR and equivalents. QSPR modeling is now widely used in the industry[4,5] as it represents fast and accurate alternatives to estimate property values as compared to other predictive tools such as equations of state or molecular simulations[6,7]. QSPR can be combined to a virtual screening procedure to highlight promising candidates for a given application within a database. This two-step method consists in estimating property values for each database's component and then filtering them according to property constraints. Such virtual screening is both restricted to the database content and to the QSPR model's boundaries.

There is still a considerable number of molecular structures unknown or not yet referenced within databases. The theoretical number of possible structures by assembling up to thirty atoms of carbon, azote, oxygen, or sulfur was estimated higher than 10^{60} by Bohacek *et al.*[8] That is, unknown but promising molecules cannot be considered during a virtual screening. Molecules' generation algorithms are one of the new pathways to help the virtual synthesis of new molecules, and both undirected and directed methodologies were developed for this purpose. Directed methodologies generate structures without targeting specific property values. For instance, it is possible to enumerate all the possible structures within a maximal number of heavy atoms – other than hydrogen[9]. Less exhaustive generation methods were also set: structures were built by assembling atoms[10] or specific fragments[11,12], and could be based on chemical reactivity[13]. Such methods tend to generate many useless structures since property values are not considered.

Directed methodologies rely on the initial compounds set to guide generations toward desired property values. The inversion of QSPR models, labeled i-QSPR hereafter, represents an emerging technique for directed generations[14]. Within i-QSPR, QSPR models are both used to predict new structures' property values and to highlight structural features relevant for molecules to possess a given specific property value. The spotted structural features, such as descriptor values or ranges to reach, can be used as constraints for the generation process[15]. Generation methods derived from Evolutionary Algorithms (EA) are also considered as directed methodologies. Genetic Algorithms (GA) are EA which modify at each iteration the so-called *parents* molecules to create *children* molecules[16–18]. Molecules are defined by a fixed number of constituents – *i.e.* fragments[16], peptides[17], ligands[18]. Only the presence and the location of constituents are modified but not their structure. This restriction limits GA-based i-QSPR to structures for which the number of constituents per structure, the available constituents, and so

the number of constituents' combinations are a priori known. When molecules cannot be defined by a fixed number of constituents, the genetic graph (GG)[19] and similar EA approaches are employed. GG works on structures at the atomic level: each molecule is considered as a graph, consisted of vertices (atoms) and edges (bonds). Crossover and mutation operators are applied to graphs and sub-elements. Constituents themselves can be modified and crossovers between structures can be performed. Globus *et al.* defined GG and showed its efficiency by optimizing a set of molecular structures until obtaining defined targeted structures by a succession of crossovers[19]. Lameijer *et al.* implemented a crossover operator and nine mutations based operators[20]. Chu and He developed the *MoleGear* software in which structures were first generated by assembling fragments and then evolved by crossovers, mutations and further fragments additions[21]. Finally, Deep-Learning (DL) techniques initially used for text or image recognition and generation were adapted to molecule generations by text strings or branched trees[14,22–25]. Nevertheless, directed methods can miss some solutions due to the bias provided by the pool of reference structures. The choice between undirected and directed methodologies to generate structures, or even the choice of the generation algorithm itself is not obvious and raises the need of tools to compare generation abilities.

Although a series of indices have been developed and widely used to compare performances of QSPR based models, the comparison of several structures' generation methods or i-QSPR methods is not obvious. Several tools are available to compare molecules between them or with respect to a target. On one hand, the similarity between two structures is computable by metrics reviewed by Nikolova *et al.*[26] and Maldonado *et al.*[27], such as the Tanimoto distance which quantifies molecules' chemical features closeness. Such tools are useful for searching for similar compounds which can exhibit similar property values within a database. On the other hand, distributions of properties and descriptors values within molecular structures sets can be

evaluated. This method is preferable to compare databases diversity. For instance, Feher and Schmidt used this approach to compare known drugs, natural molecules, and structures issued from combinatorial chemistry[28]. Brown *et al.* used both distribution comparisons and structure similarity analyses to evaluate the efficiency of several published generation algorithms implemented in the *GuacaMol* software[29]. Authors relied on the facts that generated molecules' property distributions should follow distributions within the reference database. To the author's knowledge, no tool is currently available to compare molecular generation methods without a reference database of existing compounds.

In this work, we report a new method to compare generation algorithms abilities. The exploration of the available chemical space (denoted Applicability Domain, AD) is evaluated with AD coverage, coverage unicity and representativeness indices; whilst the ability to generate molecules with specified properties values is assessed by the generation specificity index. This method is used to compare two existing molecular generation approaches, implemented and improved, based on fragment assemblies and Genetic Graphs. Additionally, molecular generation methods with and without constraints on property values were considered. So obtained methods were compared by means of new indices, for a study case: the Flash Point (FP) of hydrocarbons and oxygenated compounds. The article is organized as follows: the following section presents QSPR and i-QSPR methods, and details new indices/metrics. The subsequent section deals with the application of QSPR and i-QSPR methods to a database containing molecules with known experimental FP values. The article ends with conclusions and perspectives for this work.

METHODS

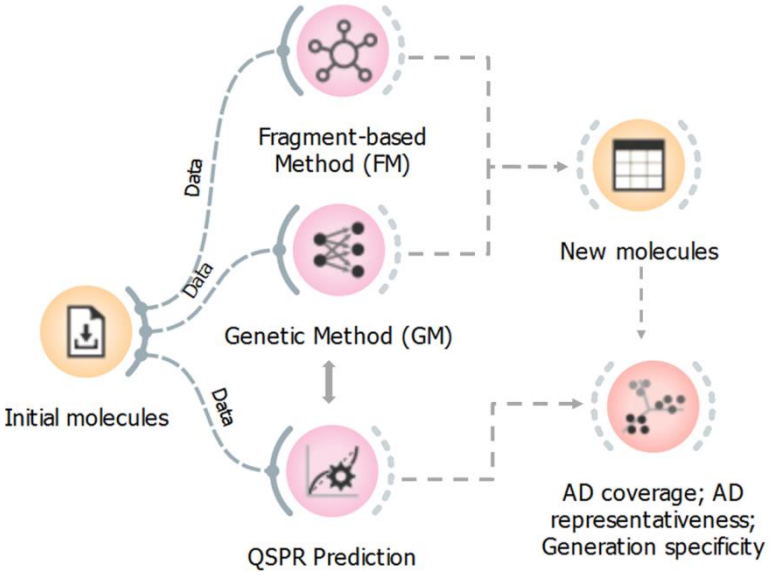

Figure 1: Molecular generation workflow as considered in this work.

The complete workflow for this work is represented in Figure 1 and can be described as follows: we start with an initial set of molecular structures for which experimental property values are known. QSPR are developed and applied to identify relevant molecular features and to predict new molecule's property values. Two algorithms are implemented and improved for molecular generation: fragments assemblies (FM, Fragment-based Methods) and successive modification of structures (GM, Genetic-based Methods). Finally, new generation methods are compared by means of new indices/metrics derived from the applicability domain coverage and representativeness.

QSPR Models

To evaluate new molecular structure relevancy for a given application, it is necessary to have accurate methods to estimate property values. We hereafter briefly describe the methodology

followed to develop QSPR models, more details can be found in reviews dealing with this topics[6,7]. The development of such models can be summarized in a 3-step procedure: (i) collecting molecular structures and their associated property values, (ii) extracting structural molecular features by means of predefined descriptors, and (iii) deriving structure-property relationships by means of machine learning (ML) techniques.

The Simplified Molecular Input Line Entry Specification (SMILES)[30] language was used to encode molecular structures to text strings[31]. A canonical SMILES format was preferred to have a unique string for each chemical structure[32]. Canonical SMILES were generated using the RDKit Python library[33].

In Silico design and Data Analysis (ISIDA) descriptors were considered to encode molecular features. ISIDA descriptors[34,35] are a series of topological fragments descriptors based on 2D Lewis graphs leading to models with good performances as shown in our previous work on surfactants properties modelling.[36] In this work, we considered all the ISIDA descriptors encoding fragments between one and four atoms resulting in 101 ISIDA descriptor spaces, corresponding to descriptors sets encoding fragments of different topologies and sizes.

The Support Vector Regression (SVR)[37] as implemented within the LibSVM library[38] was employed, with both linear and radial basis function kernels, and with an epsilon insensitive zone[39]. This method has three parameters for which the value needs to be optimized: cost, epsilon, and gamma. In the literature, parameter values to test are usually issued from random values combinations, from grid searches where each combination of parameters is tested or from optimization algorithms where the minimum number of parameters combinations is tested[40]. Here, the Sequential Quadratic Approximation (SQA)[41] method implemented in our in-house program[42] was used to optimize SVR parameter values. SQA method uses quadratic models to interpolate function surfaces without derivatives. After performing iteratively quadratic

approximations in the surrounding of several points, the function surface is estimated, and its minima are located. The procedure can lead to local minima; therefore, according to our experience the use of six starting points was sufficient to increase probabilities to reach the global minimum. Ranges of explored SVR parameters values during optimizations were in between 0.01 and 5000 for the cost values, and in between 0.0002 and 100 for both epsilon and gamma values. Optimized SVR parameters were obtained by means of an n-fold cross validation (n-CV) process[43]. n-CV randomly splits the initial dataset into n folds and uses alternatively n-1 folds (Training set) to train models and the remaining fold (Test set) to test performances of models. The Root Mean Standard Deviation (RMSD) and the coefficient of determination (R^2) indices were computed to measure models' performances. The final model was rebuilt with the best set of parameters, trained on all reference data[44]. This final model was then used to predict new compounds properties.

Applicability Domain

One important point before any application of a QSPR model is to define its Applicability Domain (AD)[45], *i.e.* the chemical space where predictions are considered as reliable. Numerous techniques were proposed for this purpose and compared in the literature[45,46]. Quantitative methods quantify the similarity between new molecules and training dataset molecules, for instance according to distance-based metrics. Such metrics can be restrictive, especially in the case where the training dataset is poor. In this work, we preferred qualitative methods which are less sensitive, even if the efficiency of QSPR cannot be quantified. We first checked AD fulfillment by performing a fragment control assessment. All possible ISIDA descriptors belonging to the modeling descriptor space were computed for new molecules. By this approach we ensured that fragments – descriptors – unknown from the initial molecules were not present

within new molecules. Structures with such fragments were rejected since the potential impact of new fragments on property values is unknown[34,46]. Then, we checked whether new molecules lied in the AD of QSPR models by a bounding box method[45]. The bounding box method first extracts from the initial dataset all descriptors' ranges of observed values and then checks that new molecules descriptor values belong to each of the associated descriptor range. The bounding box method was applied with the fragment descriptors used to model the property and the species – carbon, oxygen – within initial molecule.

Representation of the chemical space

A Principal Component Analysis (PCA)[47] was also performed on the initial data – molecules used to train the QSPR model, encoded by ISIDA descriptors with values Z-score normalized. The PCA simplified the ISIDA descriptors space dimension to principal components (PCs), each one standing in weighted linear combinations of descriptors. The first three principal components obtained an explained variance ratio of 0.37 and were used to approximate the chemical space as a 3D space. The initial molecules were then plotted in that space. We graphically approximated the initial chemical space by encompassing data points with a convex hull[45]. Since we aimed to increase small-to-medium database content by generating new structures, we requested to explore not only this initial chemical space, defined by the set of descriptors' combinations issued from the initial molecules, but all the possible descriptors' combinations. To ensure all possible generated molecule projection into the graphical chemical space, we increased each of the three axe's limit to define a parallelepiped rectangle as the new extended chemical space labeled $\mathbb{C}$. In our study, the parallelepipedic rectangle coordinates were set to be at a distance between 8 and 10 from the initial convex hull. Such coordinates allowed to project every new molecule, generated further, into that space. Note that AD fulfillment was still checked for new molecules by

fragment control assessment and by the bounding box method. Figure 2 represents the space $\mathbb{C}$, with the limits of the extended chemical space $\mathbb{C}$ drawn in green, as well as with the initial molecules projected on it in grey, and the limits of the initial convex hull drawn in violet.

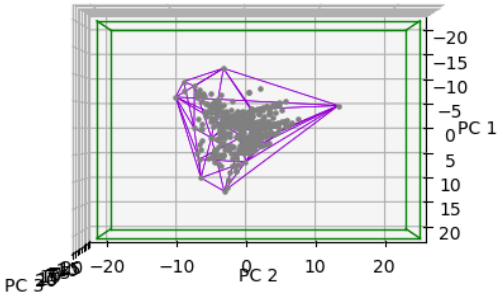

Figure 2: Representation of the space formed by the first three principal components of the PCA, with projected initial data points (grey dots), the initial chemical space (purple convex hull) and the extended chemical space $\mathbb{C}$ (green parallelepiped rectangle).

Molecules generation by fragment assemblies

The combination of fragments is a rather simple way to implement an approach to generate molecules and has already been used in the literature[11–14]. In addition, this method can be applied to a wide variety of datasets by selecting a set of fragments related to the initial molecules. We desired to avoid restricting the problem by choosing manually fragments[48], by defining their connectivity, or by setting equations[49,50]. We used then Nilakantan *et al.*[51] methodology as a basis for our method. In Nilakantan *et al.*'s work[51], molecules were constructed by assembling fragments until reaching a molecular size limit set randomly for each structure within a predefined interval. Fragments were linked together by removing hydrogen atoms and by creating

a chemical bond between fragments. Fragments' selection was weighted according to their occurrence ratio in the initial dataset.

In this study, each new molecular structure was initiated with a seed consisting of a simple carbon atom. Then, the seed was iteratively grown by randomly selecting and connecting new fragments. Each unsaturated atom was considered as a potential attachment point for another fragment. A single to triple bond was built to link fragments, according to a random choice process and atoms' valence. To avoid an over-selection of double or triple bonds as compared to the single bond, we weighted the bond type choices according to the percentage of each bond type inside our initial dataset. As in Nilakantan *et al.*'s work[51], the evolution of the molecular structure was performed either until the molecular size reached a limit chosen randomly within a range – here: the number of atoms other than hydrogen of the biggest molecule in our initial dataset –, or until the molecular structure did not possess free attachment point anymore. If the molecular size limit was reached while the structure still possessed available attachment points, hydrogen atoms were added to obtain a valid structure. Molecular descriptors' values were evaluated after each fragment addition. Structures out of the AD were downgraded, by removing the last added fragment and used as a solution without adding further fragments. Finally, generated structures uniqueness was checked; molecules already existing inside the initial dataset or already generated were discarded.

As reported in the literature, it is important to choose fragments according to the initial set of structures[11]. We first considered simple fragments: single carbon and oxygen atoms, as well as carbon-carbon and carbon-oxygen fragments with every possible bond: simple, double, triple – this method was called F0. With F1a method, fragments were extracted from ISIDA descriptors used to model the initial data by QSPR. Particularly, we selected the ISIDA descriptors defining sequences of atoms and bonds, ranging from two to four atoms[34]. These sequences were turned to

fragments by converting their typographic definition and by removing their hydrogen atoms. The random fragment choice might lead to structures having more-than-expected specific fragments, that is molecules out of the AD. We extracted then each ISIDA descriptor/fragment maximal value from the initial dataset and weighted the fragments choice according to it (this variation of F1a was labeled F1b). Table 1 proposes a summary of the investigated variations.

Table 1: Variations considered for our generation method based on fragment assemblies.

Variation	**Fragments**		**New molecules check**	
	Simples	**Dataset[1]**	**Uniq.[2]**	**AD[3]**
F0	X		X	X
F1a		X	X	X
F1b		X[4]	X	X

[1] issued from descriptors in the dataset; [2] Uniqueness: structures should not have been generated previously; [3] new structures must belong to the AD defined by fragment control assessment; [4] weighted fragments choice.

Molecules generation by iterative evolution of structures

The second investigated generation method consists in applying Genetic Graphs on a pool of molecules to design new molecular structures. At each iteration, molecules were selected randomly one-by-one. Seven different operators were considered to make structures evolve and are schematically represented on Figure 3. The crossover operation selects an additional graph from the pool of initial structures and splits the graphs into four subgraphs by removing a bond to each graph, and then link subgraphs by a single bond. Bond and atom mutations switch randomly an atom type or a bond order on a selected graph, respectively. In cyclization and decyclization operations a cycle is added or removed from molecules without fragmenting the graph. New

rings' allowed sizes were limited to five and six atoms. Fragments can be added or removed by the addition and deletion operators, respectively. At each deletion, the removed fragment was added to a library of available fragments which could be chosen further by the fragment addition operator. Parent molecules were selected and modified one-by-one, except for crossovers where two parents were selected. Valence fulfillment was checked after each modification.

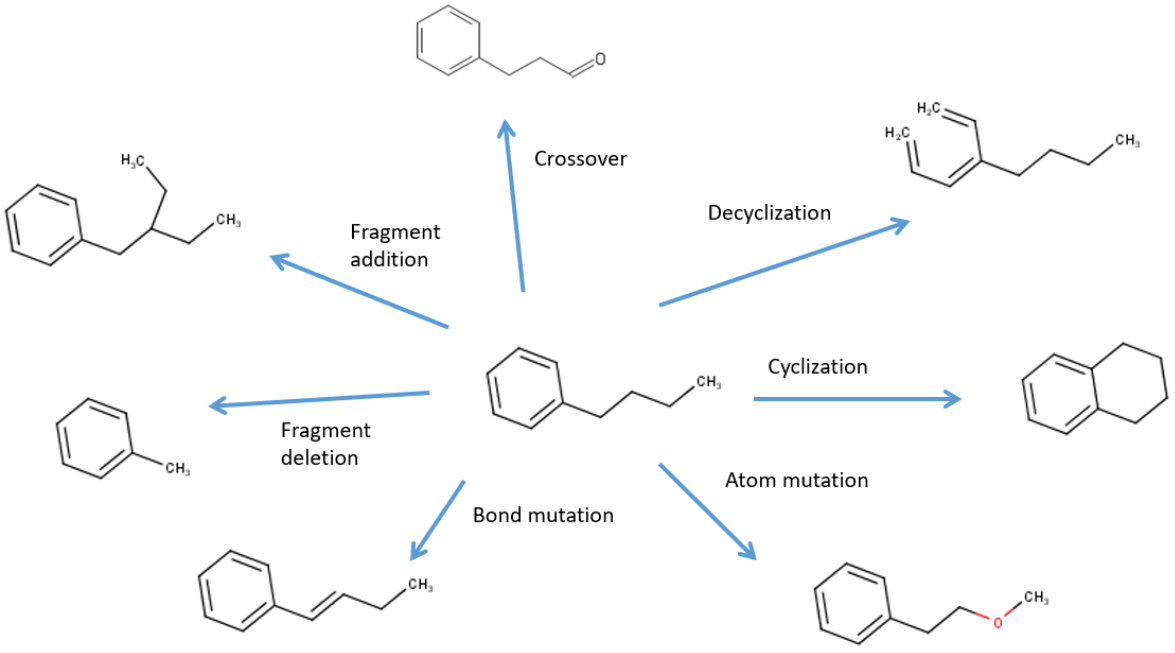

Figure 3: Considered operators for molecular structures evolution by GG.

Duplicates and molecules out of the AD were discarded. Note that when discarding a generated structure, its parent structure was processed again with the same operator until obtaining a structure satisfying constraints, or until the maximal number of attempts was reached. In this latter case and if no structure satisfying constraints was obtained, another operator was selected. If any operator allowed to generate a structure satisfying constraints after reaching their maximal number of attempts, the algorithm skipped the current *parent* on the current iteration.

The parent molecules database – initially set with molecules used to derive QSPR models – was redefined before each iteration to provide a new pool of structures to be modified. We considered two variations to supplement the pool of structures, and differences mainly stood in the management of add-ons. In the first variations (G1a and G1b) all previous parents and all generated molecules satisfying property constraints were selected as new parents for the next iteration. Only acyclic molecules were considered in G1a, without the use of the cyclization operator whilst cyclic molecules were considered in G1b. The third variation (G2) differs from G1b on the selection of molecules to modify. All initial and generated molecules were sorted according to the distance from their property's value to the targeted value, and only a specified number of the first molecules were considered as new parents. Table 2 proposes a summary of the investigated variations.

Table 2: Variations considered on our generation algorithm by successive modification of structures.

Var.	**New molecules check**		**Cycles**	**New parents' database content (selection)**
	Uniq.[1]	**AD[2]**		
G1a	X	X		All previous parents and new generated molecules.
G1b	X	X	X	All previous parents and new generated molecules.
G2	X	X	X	Previous parents and new generated molecules are sorted according to their property values closeness to the targeted value and only the top ranked molecules are selected.

[1] Uniqueness: structures should not have been generated previously; [2] new structures must belong to the AD defined by fragment control assessment.

Molecular diversity

Molecular structure generation can lead to several scenarios according to the definition of the AD and to AD's area occupancy by new structures. AD's area occupancy can vary due to chemical rules constraints (*i.e.*, octet rule fulfillment), due to algorithms, and due to user selected constraints (*i.e.*, initial structures, fragments). We therefore propose to compare generation methods considering the space formed by $\mathbb{C}$. $\mathbb{C}$ was discretized into unit cubes along the three first PC axes, and each new molecular structure is located in one of the unit cubes. As detailed hereafter, our proposed indices calculation does not consider empty cubes, limits of $\mathbb{C}$ can be increased without consequences on indices value, but this will increase computation time. We assume that sparser the molecules were projected within $\mathbb{C}$, more their structural features were diverse, since each principal component is a linear combination of descriptors. Several indices (labeled I_i) were defined according to the occupancy of each cube. Those indices were classified into three categories: AD coverage, AD representativeness, and generation specificity.

The AD coverage defines the percentage of the AD occupied by at least one molecule generated by the method *m* over the filled AD – all cubes occupied by a molecule generated by any method. In fact, some cubes could correspond to a PCA space not reachable by molecules, either due to the AD fragment control assessment or due to chemical rules' (*i.e.* octet rule) violation, and should not be considered in AD coverage. The AD coverage thus informs about the diversity of molecules generated by method *m* compared to the observed diversity of molecules generated by all methods. The I_1 index was set to inform about the AD coverage and is defined as the ratio between $N_{cubes}^{\mathbb{C},m}$, the number of occupied cubes in $\mathbb{C}$ by molecules generated by the method *m*, over $N_{cubes}^{\mathbb{C},M}$, the number of occupied cubes in $\mathbb{C}$ by molecules generated by all

methods *M* (Equation (1)). Its value is included within the interval]0;1]. I_1 values close to one indicate a high AD coverage of the generation method.

$$I_1 = \frac{N_{cubes}^{\mathbb{C},m}}{N_{cubes}^{\mathbb{C},M}}, m \in M \quad (1)$$

The AD representativeness defines how an algorithm generates molecules within the AD compared to all the generation methods. We hereafter define two occupancy rates (P_x) and use them to derive indices informing about the AD representativeness. The occupancy rate of the unit cube *x* for the method *m*, P_x^m, is defined as the ratio between $N_{structures}^{x,m}$, the number of structures generated using *m* in the unit cube *x*, and $N_{structures}^{\mathbb{C},m}$, the total number of structures generated using *m* in $\mathbb{C}$ (Equation (2)). The global occupancy rate of a unit cube *x*, P_x^T, is defined as the average (over the total number of generation methods, *M*) of individual occupancy rates P_x^m (Equation (3)).

$$P_x^m = \frac{N_{structures}^{x,m}}{N_{structures}^{\mathbb{C},m}}, x \in \mathbb{C} \quad (2)$$

$$P_x^T = \frac{\sum_{m=0}^{M} P_x^m}{M}, m \in M \quad (3)$$

Distribution indices were then used with defined P_x. Although the Kullback-Leibler divergence[52] has already been used to compare i-QSPR performances[29], this metric is not suitable for our study since P_x^m and P_x^T could be null. Therefore, four metrics are defined and tested within our work. The four metrics being similar, only I_2 is discussed within this paper; I_{2b} to I_{2d} indices are discussed in Supporting Information. I_2 is defined as one minus the Hellinger distance[53,54]. Hellinger distance is defined as the square root of the sum of the squared differences between square rooted P_x^m and P_x^T; weighted by the coefficient $1/\sqrt{2}$ to normalize I_2 values

between 0 and 1 (Equation (4)). The use of differences of square root values in I_2 gives less weight to high deviations as compared to other indices defined in Supporting Information.

$$I_2 = 1 - \left(\frac{1}{\sqrt{2}} * \sqrt{\sum_{x=0}^{n} (\sqrt{P_x^m} - \sqrt{P_x^T})^2} \right) \quad (4)$$

AD representativeness index values are within the interval [0;1]. Those indices are expected to have low values at the beginning of the generations since the number of generated structures is too low to represent algorithms performances inside the AD. Then, values should increase as soon as cubes start to be occupied by structures generated by more than one method. The higher the index values for P_x^m are, the higher its similarity to P_x^T is.

The AD coverage uniformity defines how an algorithm generates molecules in $\mathbb{C}$ compared to a hypothetical generation, providing molecules projected with an equal probability on cubes in $\mathbb{C}$. In the hypothetical generation, the hypothetical occupancy rate of a unit cube x, $P_x^{h,m}$, is defined as the ratio between $N_{structures}^{\mathbb{C},m}$ and $N_{cubes}^{\mathbb{C},m}$ (Equation (5)).

$$P_x^{h,m} = \frac{N_{structures}^{\mathbb{C},m}}{N_{cubes}^{\mathbb{C},m}}, m \in M \quad (5)$$

I_3 was set to inform about the AD coverage uniformity and is defined as one minus the Hellinger distance between P_x^m and $P_x^{h,m}$ (Equation (6)). I_3 values are within the interval [0;1]. High I_3 values indicate that the generation is producing molecules projected with an almost equal probability on the used cubes. AD coverage uniformity is similar to AD representativeness in the sense that both indices compare distributions and use a reference dynamically build. I_3 values are expected to take high values at the beginning of the generation since the algorithms start to explore $\mathbb{C}$ by generating a molecule in an unexplored cube. Then, according to algorithm implementation and to the restrictions provided by the AD, index values are expected to decrease.

$$I_3 = 1 - \left(\frac{1}{\sqrt{2}} * \sqrt{\sum_{x=0}^{n} (\sqrt{P_x^m} - \sqrt{P_x^{h,m}})^2} \right) \quad (6)$$

Finally, the generation specificity evaluates the percentage of generated molecules for given property values. I_4 is set for this purpose as the ratio between $N^m_{structures}, |p - T| \leq t$ (the number of generated molecules $N^m_{structures}$ possessing the absolute value of their property value p minus the desired property value T lower or equal to the tolerance t) and $N^m_{structures}$ (the total number of generated molecules using m) (Equation (7)). I_4 values are included within the interval [0;1].

$$I_4 = \frac{N^m_{structures}, |p - T| \leq t}{N^m_{structures}} \quad (7)$$

AD coverage, AD representativeness and AD coverage uniformity indices defined above are computed when generating molecules with or without constraints on property values. The generation specificity index is only computed when we focused on generating molecules with a desired property value. Index value comparison is valid only within the same study since the cubes' occupancy reference states are calculated in accordance with the specified generations' molecules only.

Results

Datasets and models

Flash Point (FP) is a key property for characterizing the hazardousness of chemicals. FP defines the lowest temperature for a liquid to form a mixture with the air able to ignite in the presence of a flame. Thus, Flash Point is only relevant for liquids, a low FP value reveals a high flammability risk. Many approaches have already been proposed to predict FP values of chemicals, including QSPR[6]. Correlations between the carbon atom number and FP values

highlight that large molecules tend to have high FP values[55–57]. Note that other molecular features such as branching degree and polarity also impact the FP values. For all these reasons, FP appears as an interesting case study for QSPR and i-QSPR.

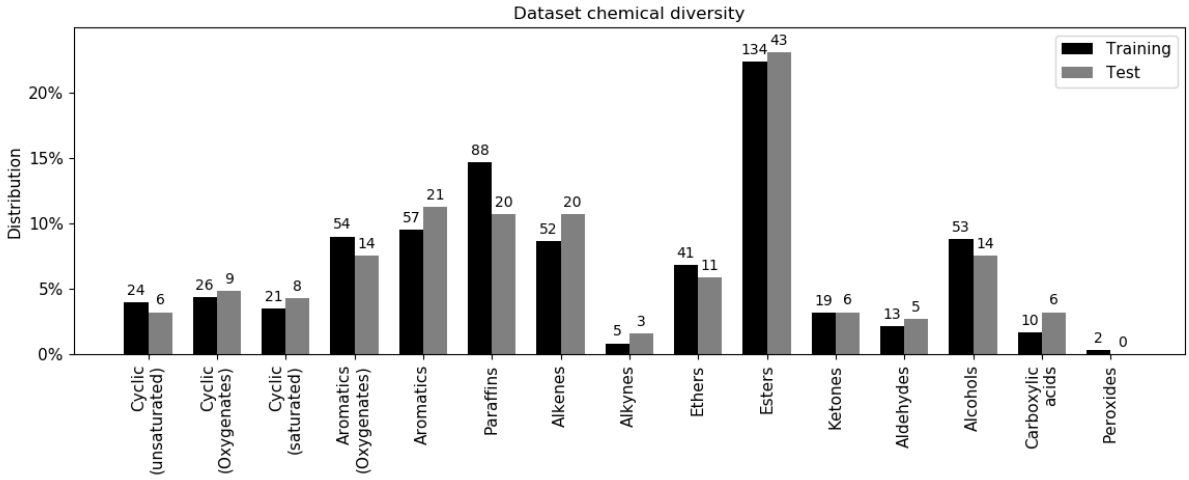

Figure 4: Composition of our dataset, in terms of percentage and number of molecules for various chemical families, in the training and test set.

We used the database previously built by Saldana *et al.*[55] Authors recovered experimental data from several sources including other QSPR studies[58–61] and databases such as Design Institute for Physical Properties (DIPPR)[62]. In the work by Saldana *et al.*, the database was filtered to keep only compounds of interest, for instance hydrocarbons and oxygenated molecules (mainly alcohols and esters). We considered in our work the full database, including additional families of compounds such as aldehydes, ketones, ethers, and alkynes. This database that contains 785 chemicals was split randomly into two subsets, 599 compounds were used for training and 186 for testing the models. Figure 4 illustrates the chemical diversity within the database. Each training and test set contains about 40% hydrocarbons and 60% of oxygenated compounds. Among them, 30% (240 molecules) are cyclic, including naphthenes and aromatics. The most

represented chemical family is the esters with 177 molecules (21 %) while there are few alkynes (8 compounds) and peroxides (2 compounds).

The ISIDA descriptors were computed to encode molecular features on the basis of SMILES. For each descriptor set, parameters of SVR were optimized using 5-CV according to the methodology described above. Models using descriptors based on sequences of two to four atoms and their bonds exhibits good performances as shown in Table 3, according to internal (cross-validation) and external validation. When predicting Saldana *et al.* dataset,[55] our model shows performances similar to those reported by authors. The slight difference in performances can be attributed to our database which contains a wider diversity than that used by Saldana *et al.* and to the use of a single QSPR whilst Saldana *et al.* used several QSPRs into a consensus model. An analysis of the accuracy of our QSPR model within the space $\mathbb{C}$ is proposed in Supporting Information (Figure S3), it shows that the model prediction error is roughly stable in this space.

Table 3: QSPR model performances measured by cross-validation and external validations, compared to that reported by Saldana *et al.*[55]

Dataset	**This work's performances**		**Previous performances**	
	RMSD (K)	**R^2**	**RMSD (K)**	**R^2**
Training set (cross-validation)	15.74	0.920	---	---
External validation set	15.46	0.935	---	---
Saldana *et al.*'s full dataset[55]	12.71	0.948	10.9	0.959
Saldana *et al.*'s validation set[55]	9.58	0.967	10.9	0.967

Our dataset's molecules are provided with their SMILES notation, their predicted FP value, and their descriptors values in Supporting Information. The maximal occurrence of each fragment can be found by searching the maximal value of its associated descriptor. The QSPR AD is restricted by those chemical features known within the initial set and encodable by the descriptors used. For instance, up to one peroxide group is allowed within new molecules (as three initial molecules contain this feature in the used database), as well as one ethynyl group (included within eight initial molecules). Bigger fragments, such as ketal group, cannot be encoded by 4-atoms fragments and their presence is therefore not checked. Also, the hybridization of carbon atoms within molecules is not fully considered by the used descriptors.

Generation of diverse molecules

The dataset was then used as the reference for molecular generations. New structures were built by fragment assemblies (methods F0, F1a and F1b) and iterative evolution of a population of structures (methods G1a and G1b). Up to 5 million structures were generated with each method. Noting that such number of structures is sufficient for a first evaluation of methods. Fragment-based methods (FMs) can output in a single run all news structures. GM methods must be performed in several runs. In fact, GMs' initial pool of structures is the QSPR dataset – with or without cyclic molecules according to G1a or G1b variation. After a 10-iteration run, this pool contains more than $8x10^5$ structures and quickly become difficult to handle by our computer's memory. Four runs were needed for G1a and G1b to reach a total of 5 million generated molecules after removing duplicates.

For all generated structures as well as for molecules of the initial dataset, we computed FP values and a series of characteristics such as the molecular weight (MW)[41] and synthetic accessibility (SA)[63]. The molecular weight informs about molecule's size which is correlated to

FP[55–57], and SA scores compounds from 1 (easy to synthesize) to 10 (very complicated to synthesize). SA is a useful property since we aim to propose synthetically feasible compounds as solutions. The distributions of each property values according to the generation method employed are represented on Figure 5. The molecular weight distribution is right skewed for generated and initial sets. Initial molecules' MW distribution appears to be bimodal with the highest peak located around 120 g/mol. MW distributions of generated molecules are monomodal with their peaks at roughly 200 g/mol to 250 g/mol. F1b's MW values are more spread than for the other generations. Also, the SA distribution for initial molecule is spread between 0.5 and 5.5 whilst generated molecules' SA distribution is spread between 1.5 and 6 with a peak around 4 (for molecules generated by G1a/b) and around 4.5 (for molecules generated by F0/F1/F2a). Predicted FP values for initial molecules are spread between 200 K and 500 K whilst FP distributions of generated molecules are narrower and centered around 400 K. As expected, generated molecules tend to be bigger than the average of the initial molecules, also to be more complex and thus to possess a high FP value. That phenomenon is observed since the number of possible atoms combinations to generate molecules with low FP value – small molecules – is smaller than for structures with a higher FP value. It should be noted that such generated big molecules still belong to the AD defined by a fragment control assessment and bounding box. Molecules generated by G1a and G1b methods seem to be slightly easier to synthesize according to their SA values and molecules generated by F1b to be sparser about their size according to their molecular weight.

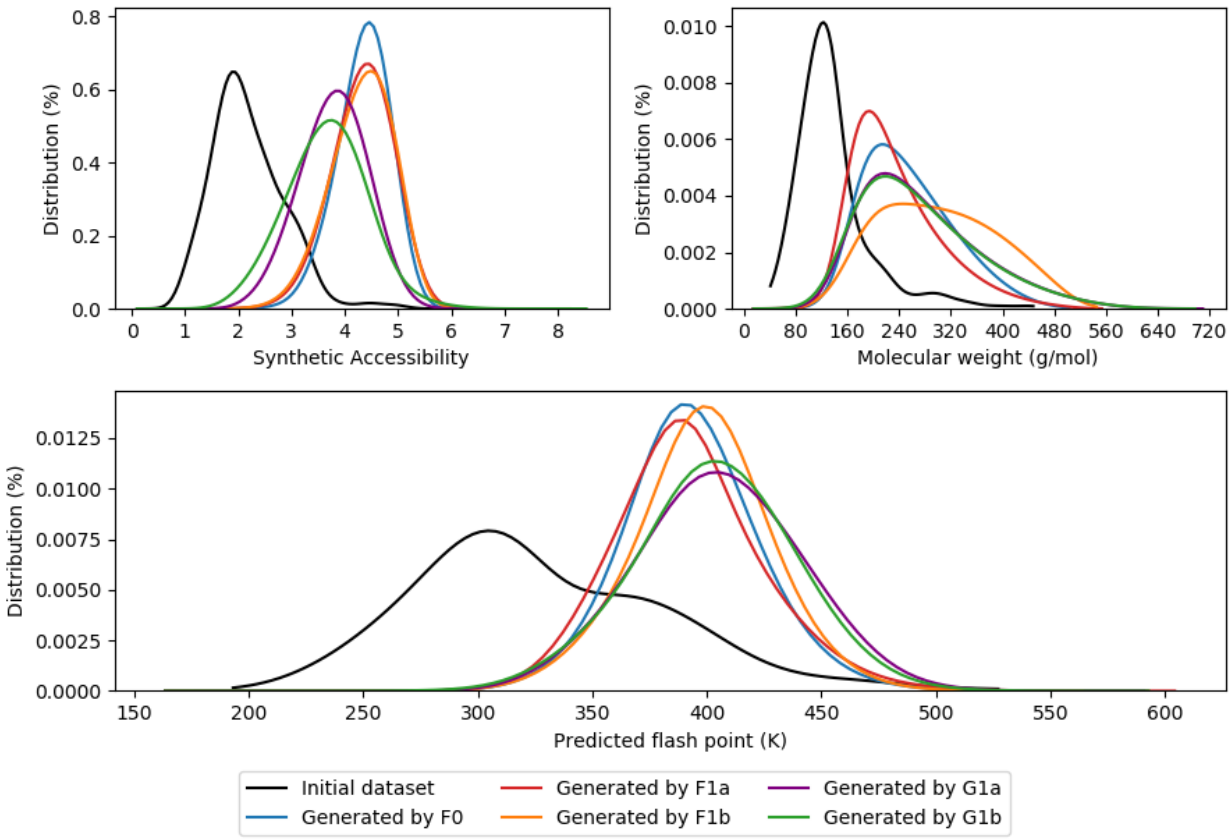

Figure 5: Synthetic accessibility, molecular weight, and flash point distributions (percentage of molecules from the considered dataset) within sets of initial and generated molecules, for each generation method.

Several cube sizes from 0.5 to 6 unit were investigated to discretize the space $\mathbb{C}$. A cube size of 1 unit exhibited the best compromise between the stability of indices values and the required computing time. $\mathbb{C}$ was thus discretized into about 40^3 cubes. Indices I_1 to I_3 were dynamically computed for each investigated generation method. Table 4 presents the indices calculated for the 5 million generated molecules.

Table 4: Calculated indices after generation of five million molecules with each method.

Method	I_1	I_2	I_3
F0	0.68	0.77	0.38
F1a	0.74	0.62	0.41
F1b	0.72	0.69	0.40
G1a	0.75	0.79	0.43
G1b	0.90	0.84	0.44

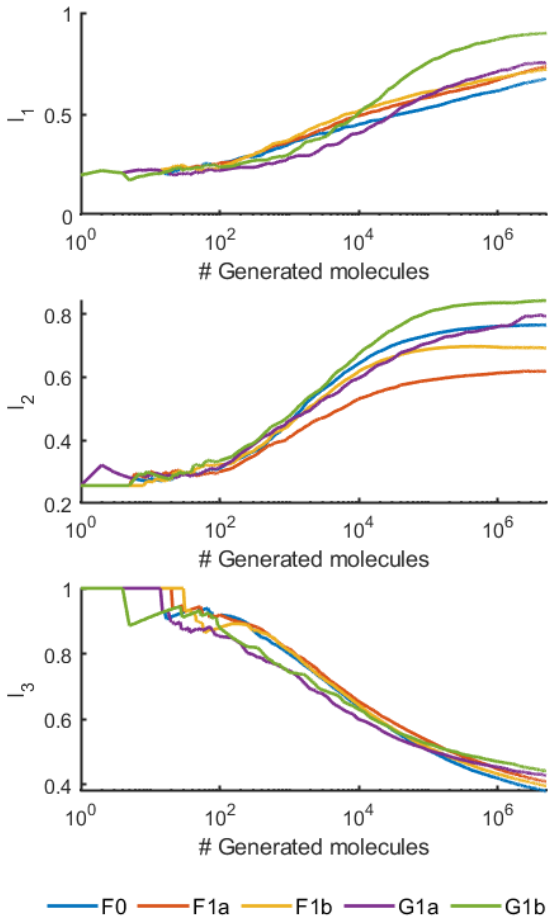

Figure 6: Evolution of indices I_1 to I_3 with the number of generated molecules, for each generation method.

Figure 6 presents the variation of indices I_1 to I_3 as a function of the number of generated molecules for the different generation methods. At the beginning of the generations, I_1 values are low around 0.2, *i.e.* around 1/M with M=5 the number of compared methods, as each of the five methods provided molecules located in different cubes. After about one hundred generated molecules, I_1 values start to increase, meaning that areas of individual generations start to overlap. I_1 values of FMs tend to inch up faster than GMs. Among FMs, F1a and F1b which use more fragments as compared to F0 were able to explore wider $\mathbb{C}$. Concerning GMs, their I_1 values also increase but more slowly than FMs before 10^4 generated molecules: GMs are restricted by the diversity within the pool of initial structures. After first iterations – 10^4 to 10^5 generated molecules –, GMs outperform FMs in terms of AD coverage. Finally, at five million generated structures, the percentage of occupied cubes is higher for GMs as compared to FMs. Since covering the maximum of $\mathbb{C}$ is a key point to provide diverse generated molecules, GMs score better than FMs from 10^5 generated structures.

Variations of AD representativeness's index I_2 with the number of generated structures can be also observed on Figure 6. At the beginning of the generations, values are at their minimum, meaning an important deviation between distributions of generated molecules by each method and the distribution of molecules generated by all methods. Then, values increase with the number of generated structures, showing that each method started to generate structures projected in a more similar way in $\mathbb{C}$. Plateau values are observed from 10^5 generated structures. Values of AD representativeness's index I_2 at five million generated structures are reported on Table 4. I_2 index ranks generation methods as follows: G1b>G1a>F0>F1b>F1a.

AD coverage uniformity index has its values close to one at the beginning of the generations, meaning that as expected the same number of generated molecules is projected in each cube of $\mathbb{C}$.

I_3 index values start then to decrease with the number of generated molecules. Indeed, as soon as algorithms start to generate molecules in already explored cubes, the distribution of generated molecules in cubes start to differ from a uniform distribution. At 5 million generated molecules, I_3 rank generation methods as follow: G1b>G1a>F1a>F1b>F0.

Fragments constraints imposed F1a and F1b have tendencies to lead to the generation of molecules in more dissimilar way as compared to F0 or to GMs. According to those observations, we conclude to a better efficiency for GMs to generate molecules in comparison with FMs. Indeed, from comparisons performed in this section, GMs appear to be interesting methods providing both the closest AD coverage and the highest AD representativeness. The consideration of rings in molecules and genetic operators as in G1b, improves the molecular diversity as compared to G1a.

Generation of diverse molecules within a desired property value range

In this section, we investigate the generation of structures with predicted property values within a specific interval. Three FP ranges were arbitrary selected: [200 K, 300 K[, [300 K, 400 K[, and [400 K, 500 K[. The targeted FP ranges were seen as an average value – 250 K, 350 K and 450 K – associated with a tolerance value of 50 K. From conclusions drawn in previous sections, we hereafter employed GMs: G1b and G2. With G2, we varied the number of parents between 50 and 800 and we present hereafter the results when the generation was restricted to 800 molecules (*i.e.*, close to the initial set size; this generation is labeled G2-800) and to 50 molecules (labeled G2-50). Due to the lack of selected structures' diversity, the ability of G2 methods to output new structures quickly decreases with the number of iterations. Therefore to obtain five million unique structures, generations had to be performed up to five times with G2-800 for each set of targeted property value range, and up to 100 times with G2-50. Duplicated molecules between

two runs of the same method were removed. Since each run generated new structures independently, the pool of selected structures to be modified at the end of each iteration was different from one run to another; inducing an exploration of different chemical patterns over runs.

Table 5: Generation method performances for molecules with a FP value included in each interval.

Targeted interval	[200 K, 300 K[				[300 K, 400 K[				[400 K, 500 K[			
#molecules	3.53 x 10^3				1.48 x 10^6				2.75 x 10^6			
Index / Method	I_1	I_2	I_3	I_4[a]	I_1	I_2	I_3	I_4	I_1	I_2	I_3	I_4
G1b	0.82	0.86	0.55	2.35	0.94	0.77	0.44	0.45	0.96	0.80	0.44	0.55
G2-50	0.82	0.85	0.54	0.07	0.73	0.80	0.41	0.30	0.66	0.81	0.45	0.71
G2-800	0.80	0.85	0.55	0.37	0.81	0.88	0.40	0.36	0.85	0.90	0.43	0.62

[a] x10^{-2}

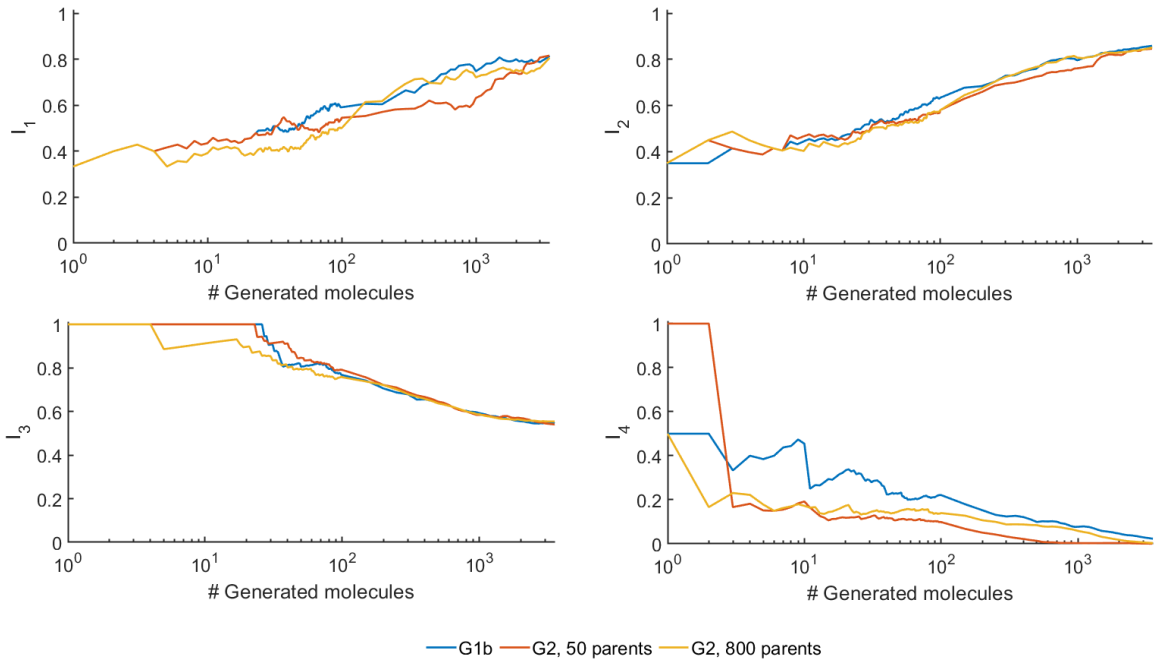

Figure 7: Indices variations as a function of the number of molecules generated having their FP value in the interval [200 K, 300 K[.

Each method provided a minimum of 3.53 x 10^3 structures having their predicted FP value in [200 K, 300 K[among the 5 million generated structures. Index values after generating 3.53 x 10^3 structures with FP values within the specified range are reported in Table 5. Evolution of indices with the number of generated structures respecting the FP requirement are shown on Figure 7. AD coverage, AD coverage uniformity and representativeness indices' values are similar for each considered method; showing similar distributions of molecules generated by each method. According to generation specificity, generating molecules with FP values in the interval [200;300[was easier with G1b, when the pool of structures to modify was not restrained, than with G2, when selecting the structures to modify at each iteration.

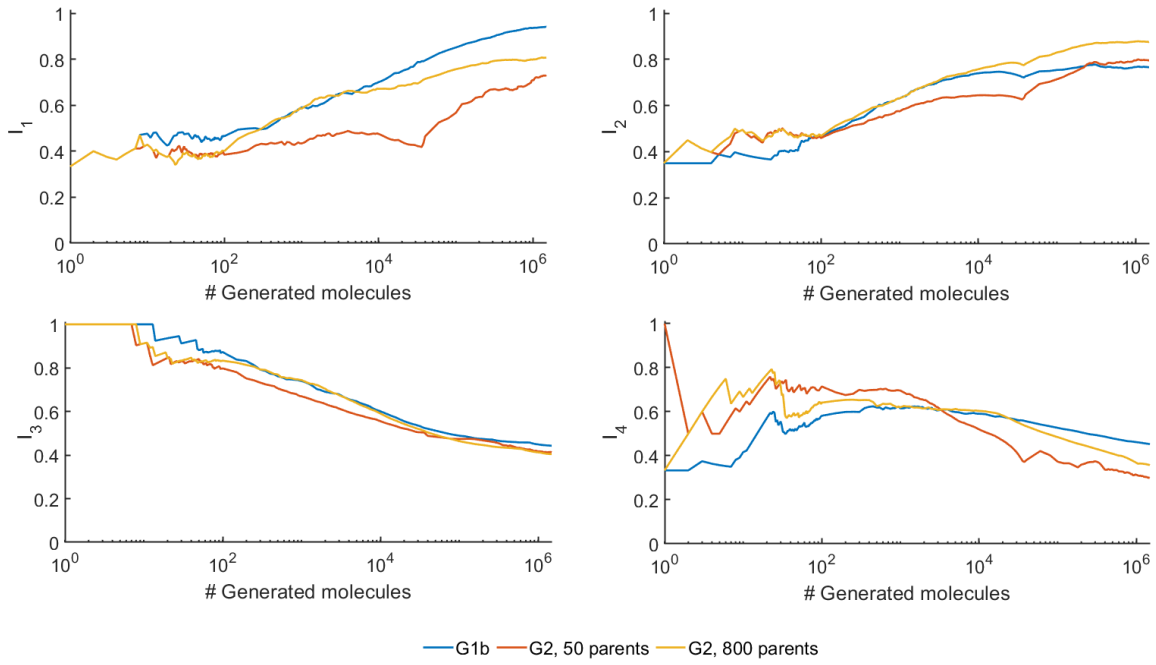

Figure 8: Indices variations as a function of the number of molecules generated having their FP value in the interval [300 K, 400 K[.

Each considered GM was able to provide a minimum of 1.48 million structures possessing their FP value in the second interval [300 K, 400 K[among the 5 million structures generated. Index values after generating 1.48 million structures with FP values within the specified range are reported in Table 5. Evolution of indices with the number of generated structures respecting the FP requirement are presented on Figure 8. From roughly $1x10^4$ to $4x10^4$ generated structures, G2-50 I_1, I_2 and I_4 values decrease and then increase again: this behavior shows the transition between two G2-50 generation runs. From 10^4 generated structures, G1b leads to the best AD coverage, followed by G2-800 and G2-50. The best AD representativeness is obtained at the end of the generations by G2-800, followed by G2-50's and G1b's. G1b obtains a slightly better AD coverage unicity than G2-800 and G2-50. Regarding the generation specificity, I_4 values increase at the beginning of the generation process until 10^3 structures, where a decrease is observed for

the three methods. G1b most easily generated molecules within the second targeted FP interval, followed by G2-800 and G2-50.

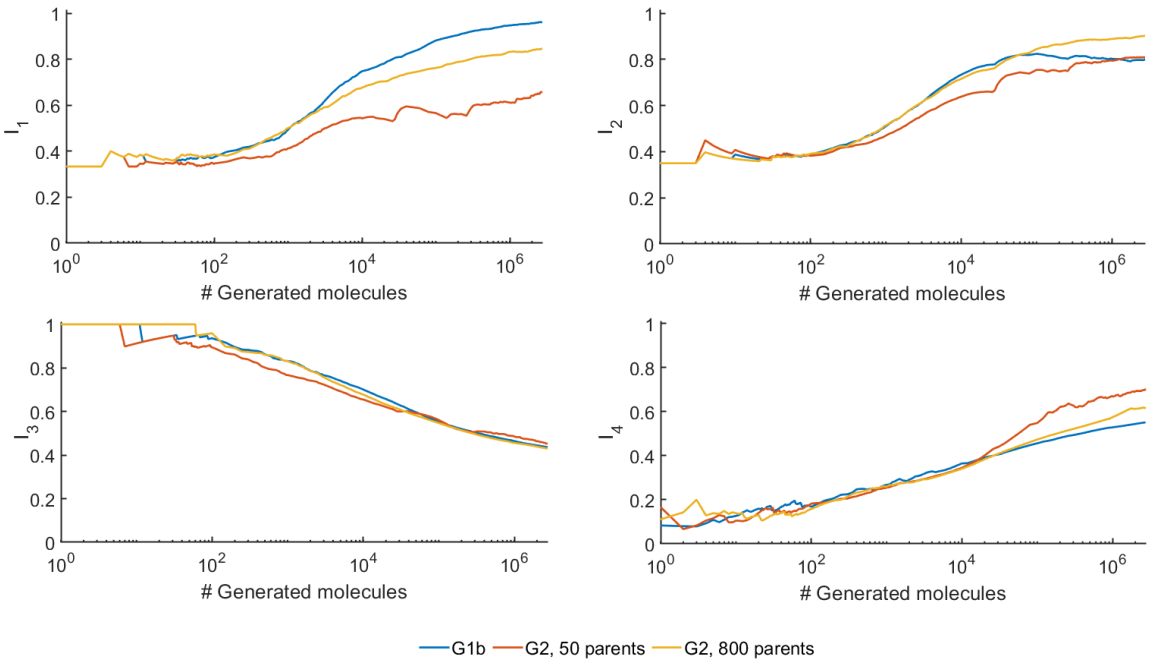

Figure 9: Indices variations as a function of the number of molecules generated having their FP value in the interval [400 K, 500 K[.

Each method generated a minimum of 2.75 million structures with a predicted FP in the third interval [400 K, 500 K[among the 5 million structures generated. Index values after generating 2.75 million structures with FP values within the specified range are reported in Table 5. Evolution of indices with the number of generated structures respecting the FP requirement are presented on Figure 9. According to I_1 values, the best AD coverage is obtained with the use of G1b, G2-800 and finally G2-50. On one side, we notice a smooth decrease of the AD representativeness index values for G1b's method from $4x10^4$ generated structures. At the opposite side, G2-800 and G2-50 index values continue increasing. At the end of the generations, G2-800 obtains the best AD representativeness scores, followed by G2-50 and G1b. The AD

coverage unicity is similar for each compared method. I_4 values increase with the number of generated structures, showing an increasing ability to generate molecules satisfying property constraints for all methods. At the end of generations, G2-50 obtains the best generation specificity, followed by G2-800 and G1b.

For each targeted interval, AD coverage, and representativeness index values according to the number of generated structures follow the same behavior as when property values are not targeted: index values increase as soon as some cubes start to be occupied by molecules generated by more than one method. The use of several runs for each method allowed to smooth the lack of diversity within the pool of structures to be processed, this is especially the case with G2-50. The study of the AD coverage highlights a better efficiency for the G1b method to explore the available AD. This difference in AD coverage induces a stabilization or a decrease of G1b AD representativeness values, as observed after 10^4 generated molecules. The AD coverage unicity values follow the same behavior as when property values are not targeted. This shows that generated molecules are not evenly spread on the occupied space in $\mathbb{C}$. Values are however similar between the generations for each targeted interval, indicating a similar spreading of molecules. Concerning generation specificity, we observe for the first two intervals an increase of I_4 values until 10^2 (for the first FP interval) or 10^3 (for the second interval) generated molecules, followed by a decrease of those values. To generate structures with those FP requirements become harder after a threshold related to the mean expected size of molecules. The combinatorial being higher for the third interval as compared to first and second, an increase of I_4 values is observed with increasing number of generated molecules.

Overall the targeted intervals, G1b provides high AD coverage scores, and so more diverse molecules than G2. This behavior allows G1b to spot more easily new structural features required

for small to medium molecules to possess a desired property, and so to generate more easily small to medium molecules. G2 methods reduce the diversity by focusing on specific patterns in the selection procedure. G2 is better suited for the generation of large molecules since the number of allowed structural modifications is high with such molecules. This behavior is especially noticed when using a small number of selected items per iteration, *i.e.* with G2-50. GMs could be improved further to obtain even better AD coverage and AD representativeness, for instance by adding more operators to handle cycles within aromatics and naphthenes, like the ring mergence recently proposed by Inoue *et al.*[64]

Conclusion

We addressed the problem of molecular generation and i-QSPR by revisiting two known approaches: generation of molecules by Fragments-based Methods (FM) and Genetic-based Methods (GM). FM uses fragments which correspond either to carbon and/or oxygen atoms linked by a bond or which are functional groups extracted from the initial pool of structures. GM successively modifies an initial pool of structures by means of genetic graphs operators which act on atoms and bonds. In order to compare generation methods, we proposed a series of indices based on a discretization into unit cubes of the chemical space $\mathbb{C}$. Those indices analyze the pool of generated structures in terms of (i) AD coverage, (ii) AD representativeness, (iii), AD coverage unicity, and (iv) generation specificity when a property's value is targeted.

FM and GM-based methods were implemented and compared with these new indices when generating molecules for the Flash Point (FP) endpoint. We first concluded on the better efficiency of GMs as compared to FMs to output structures projected sparser in $\mathbb{C}$. In fact, GMs have a wider choice of tools to generate structures – several operators – than FMs which can only add fragments. Moreover, considering only fragment additions, FMs use a fixed number of

defined fragments whilst GMs use fragments dynamically issued from initial and generated molecules. Also, GMs handle cyclic molecules whilst this feature was not encoded for FMs. Then, we focused on GMs to generate molecules having their FP values within three desired intervals. The method G1b allowed to generate molecules sparser distributed in $\mathbb{C}$ and so to ease the production of small-to-medium structures for which the number of atoms combinations is limited. G2 methods performed better than G1b to generate bigger structures. We would like to emphasize that conclusions drawn in this study were performed based on one database including FP values. Future works will deal with extensive comparisons between FMs and GMs using the proposed indices, testing their behaviors on many other databases varying the initial pool of molecular structures and target properties.

This method was presented for the case where the applicability domain (AD) of the QSPR model was used both to check new molecules similarity to the initial data and to define a chemical space for comparing generation methods. It is possible to restrict more the AD by considering ranges of additional fragments (for instance longer or using another topology) as constrains. Moreover, it is possible to define a generation space manually, i.e., a subset of interest within the AD, to compute the proposed indices. For instance, within this study, when targeting a flash point range, it is possible to only consider unit cubes or in their surrounding in $\mathbb{C}$ occupied by initial molecules having their property within the desired range. We did not use this feature because it could bias the exploration of the whole chemical space.

The proposed indices use a chemical space representation built by PCA according to the initial database. This allows them to compare generation methods without considering an external reference database. Indeed, the comparisons' reference is dynamically built from the generated items. Authors would like to highlight that index values are related to the number of generated structures as well as to the compared methods. Moreover, as the reference is dynamically defined,

comparisons can only be performed for a same number of generated molecules. Index values are dependent on the simplification of the descriptors' space by the PCA and three dimensions are often not sufficient to explain the full variance in the dataset. The use of other methods to represent chemical spaces, such as with the BCUT metrics[65], or with 2-Dimension data representations by Generative Topographic Mapping[66,67], are under investigation.

Our study can also be extended to the optimization of molecules fulfilling constrains on several properties with the following adaptations. First, individuals QSPR models should be built for each property according to initial dataset(s). Then, a global applicability domain should be defined, considering only the chemical features represented in all initial dataset(s). The scoring function of generated molecules should consider all properties, it could be defined as the mean of scaled differences between predicted and desired properties values. The procedure to build $\mathbb{C}$ according to the global AD and to compute index values remains unchanged.

AUTHOR INFORMATION

Corresponding Author

*E-mail: benoit.creton@ifpen.fr

ACKNOWLEDGMENT

The authors are grateful to A. Varnek research team for discussions about distribution comparisons, and to D. Sinoquet for discussions about the use of the SQA optimization algorithm.

ABBREVIATIONS

QSPR, Quantitative Structure-Properties Relationship; i-QSPR, inversion of QSPR model; GA, Genetic Algorithm; GG, Genetic Graphs; AD, Applicability Domain; FM, generation by fragments assemblies; GM, generation by successive modification of structures; SVR, Support Vector Regression; SQA, Sequential Quadratic Approximation; n-CV, n-fold Cross Validation; RMSD, Root Mean Standard Deviation; PCA, Principal Component Analysis; FP, Flash Point; SMILES, Simplified Molecular Input Line Entry Specification; MW, Molecular Weight; SA, Synthetic Accessibility.

SUPPORTING INFORMATION AVAILABLE: Initial dataset of molecules, with their descriptors values and their predicted flash point; Additional information about the considered PCA space and extended chemical space $\mathbb{C}$; Some of the AD representativity indices I_{2b}, I_{2c} and I_{2d} definition, application, and comparisons with I_2; Analysis of QSPR model's accuracy within the space $\mathbb{C}$.

DATA AND SOFTWARES AVAILABILITY

DATA: The molecules used to build the QSPR model and as basis for generations are provided with their SMILES notation, their predicted FP value, and their descriptors values in Supporting Information.

QSPR MODELS: Descriptors were calculated by the *ISIDA Fragmentor 2017* software (available on request at http://infochim.u-strasbg.fr/spip.php?rubrique41). The construction of models was handled by an in-house bash script, using the *libsvm* software[44] (available on

https://www.csie.ntu.edu.tw/~cjlin/libsvm/), and parameter values were optimized by the software reported in Sinoquet *et al.*[48] paper. Predictions were performed by another in-house bash script, using also *ISIDA Fragmentor 2017* and *libsvm*.

GENERATION ALGORITHMS: Generations were proceeded by in-house python (version 3.6.6) scripts, using the following libraries: *RDKit*[39] for the manipulation of molecules (available on https://www.rdkit.org/, version 2018.09.3), and *Numpy* (available on https://numpy.org/, version 1.16.3). Also, the following standard python libraries were used: *multiprocessing* to process several molecules at the same time, *subprocess* to predict molecules' properties by the bash script mentioned above, *csv* to read input data and write results.

COMPARISON ALGORITHMS: Molecules were compared with python (version 3.6.6) scripts using the following libraries: *sklearn*[68] for the standardization of descriptors and to compute PCAs (available on https://scikit-learn.org, version 0.23.2), *scipy*[69] to compute convex hulls and space $\mathbb{C}$ (available on https://www.scipy.org/, version 1.3.0), *matplotlib*[70] to graphically represent convex hulls and space $\mathbb{C}$ (available on https://matplotlib.org/, version 3.0.0), *numpy* to compute index values (available on https://numpy.org/, version 1.16.3). Also, the following standard python library was used: *csv* to read input data and write results.

REFERENCES

(1) Apodaca, R. L. *Sixty-Four Free Chemistry Databases*. https://depth-first.com/articles/2011/10/12/sixty-four-free-chemistry-databases/ (accessed 2021-04-26).

(2) Lin, A.; Horvath, D.; Afonina, V.; Marcou, G.; Reymond, J.-L.; Varnek, A. Mapping of the Available Chemical Space versus the Chemical Universe of Lead-Like Compounds. *ChemMedChem* **2018**, *13*, 540–554. DOI: 10.1002/cmdc.201700561.

(3) Varnek, A.; Baskin, I. Machine Learning Methods for Property Prediction in Chemoinformatics: Quo Vadis? *J. Chem. Inf. Model.* **2012**, *52*, 1413–1437. DOI: 10.1021/ci200409x.

(4) Westmoreland, P.; Kollman, P.; Chaka, A.; Cummings, P.; Morokuma, K.; Neurock, M.; Stechel, E.; Vashishta, P. *Applications of Molecular and Materials Modeling*, International Technology Research Institute, Baltimore, 2002.

(5) Cherkasov, A.; Muratov, E. N.; Fourches, D.; Varnek, A.; Baskin, I. I.; Cronin, M.; Dearden, J.; Gramatica, P.; Martin, Y. C.; Todeschini, R.; Consonni, V.; Kuz'min, V. E.; Cramer, R.; Benigni, R.; Yang, C.; Rathman, J.; Terfloth, L.; Gasteiger, J.; Richard, A.; Tropsha, A. QSAR Modeling: Where Have You Been? Where Are You Going To? *J. Med. Chem.* **2014**, *57*, 4977–5010. DOI: 10.1021/jm4004285.

(6) Nieto-Draghi, C.; Fayet, G.; Creton, B.; Rozanska, X.; Rotureau, P.; Hemptinne, J.-C. de; Ungerer, P.; Rousseau, B.; Adamo, C. A General Guidebook for the Theoretical Prediction of Physicochemical Properties of Chemicals for Regulatory Purposes. *Chem. Rev.* **2015**, *115*, 13093–13164. DOI: 10.1021/acs.chemrev.5b00215.

(7) Katritzky, A. R.; Kuanar, M.; Slavov, S.; Hall, C. D.; Karelson, M.; Kahn, I.; Dobchev, D. A. Quantitative Correlation of Physical and Chemical Properties with Chemical Structure:Utility for Prediction. *Chem. Rev.* **2010**, *110*, 5714–5789. DOI: 10.1021/cr900238d.

(8) Bohacek, R. S.; McMartin, C.; Guida, W. C. The art and practice of structure-based drug design: A molecular modeling perspective:A Molecular Modeling Perspective. *Med. Res. Rev.* **1996**, *16*, 3–50. DOI: 10.1002/(SICI)1098-1128(199601)16:1<3:AID-MED1>3.0.CO;2-6.

(9) Ruddigkeit, L.; van Deursen, R.; Blum, L. C.; Reymond, J.-L. Enumeration of 166 billion Organic Small Molecules in the Chemical Universe Database GDB-17. *J. Chem. Inf. Model.* **2012**, *52*, 2864–2875. DOI: 10.1021/ci300415d.

(10) Hoksza, D.; Skoda, P.; Voršilák, M.; Svozil, D. Molpher: a Software Framework for Systematic Chemical Space Exploration. *J. Cheminformat.* **2014**, *6*, 7. DOI: 10.1186/1758-2946-6-7.

(11) Gani, R.; Brignole, E. A. Molecular Design of Solvents for Liquid Extraction Based on UNIFAC. *Fluid Phase Equilib.* **1983**, *13*, 331–340. DOI: 10.1016/0378-3812(83)80104-6.

(12) Brignole, E. A.; Bottini, S. B.; Gani, R. A Strategy for the Design and Selection of Solvents for Separation Processes. *Fluid Phase Equilib.* **1986**, *29*, 125–132. DOI: 10.1016/0378-3812(86)85016-6.

(13) Hartenfeller, M.; Zettl, H.; Walter, M.; Rupp, M.; Reisen, F.; Proschak, E.; Weggen, S.; Stark, H.; Schneider, G. DOGS:Reaction-Driven de Novo Design of Bioactive Compounds. *PLoS Comput. Biol.* **2012**, *8*, e1002380. DOI: 10.1371/journal.pcbi.1002380.

(14) Gantzer, P.; Creton, B.; Nieto-Draghi, C. Inverse-QSPR for De Novo Design:A Review. *Mol. Inf.* **2020**, *39*, 1900087. DOI: 10.1002/minf.201900087.

(15) Miyao, T.; Kaneko, H.; Funatsu, K. Ring-System-Based Exhaustive Structure Generation for Inverse-QSPR/QSAR. *Mol. Inf.* **2014**, *33*, 764–778. DOI: 10.1002/minf.201400072.

(16) Devi, R. V.; Sathya, S. S.; Coumar, M. S.; Selvaraj, M. GAMol: Genetic Algorithm Based De Novo Molecule Generator. In *International Conference on "Advances in Computing, Communication and Information Science"*: Kollam, Kerala, 2014.

(17) Cho, S. J.; Zheng, W.; Tropsha, A. Rational Combinatorial Library Design. 2. Rational Design of Targeted Combinatorial Peptide Libraries Using Chemical Similarity Probe and the Inverse QSAR Approaches. *J. Chem. Inf. Comput. Sci.* **1998**, *38*, 259–268. DOI: 10.1021/ci9700945.

(18) Fan, C.; Springborg, M.; Feng, Y. Application of an Inverse-Design Method to Optimizing Porphyrins in Dye-Sensitized Solar Cells. *Phys. Chem. Chem. Phys.* **2019**, *21*, 5834–5844. DOI: 10.1039/c8cp07722c.

(19) Globus, A.; Lawton, J.; Wipke, T. Automatic Molecular Design Using Evolutionary Techniques. *Nanotechnology* **1999**, *10*, 290. DOI: 10.1088/0957-4484/10/3/312.

(20) Lameijer, E.-W.; Kok, J. N.; Bäck, T.; Ijzerman, A. P. The Molecule Evoluator. An Interactive Evolutionary Algorithm for the Design of Drug-Like Molecules. *J. Chem. Inf. Model.* **2006**, *46*, 545–552. DOI: 10.1021/ci050369d.

(21) Chu, Y.; He, X. MoleGear:A Java-Based Platform for Evolutionary De Novo Molecular Design. *Molecules* **2019**, *24*, 1444. DOI: 10.3390/molecules24071444.

(22) Gómez-Bombarelli, R.; Wei, J. N.; Duvenaud, D.; Hernández-Lobato, J. M.; Sanchez-Lengeling, B.; Sheberla, D.; Aguilera-Iparraguirre, J.; Hirzel, T. D.; Adams, R. P.; Aspuru-Guzik, A. Automatic Chemical Design Using a Data-Driven Continuous Representation of Molecules. *ACS Cent. Sci.* **2018**, *4*, 268–276. DOI: 10.1021/acscentsci.7b00572.

(23) Guimaraes, G. L.; Sanchez-Lengeling, B.; Outeiral, C.; Farias, P. L. C.; Aspuru-Guzik, A. *Objective-Reinforced Generative Adversarial Networks (ORGAN) for Sequence Generation Models*. https://arxiv.org/abs/1705.10843v3 (accessed 2021-04-26).

(24) Winter, R.; Montanari, F.; Steffen, A.; Briem, H.; Noé, F.; Clevert, D.-A. Efficient Multi-Objective Molecular Optimization in a Continuous Latent Space. *Chemical science* **2019**, *10*, 8016–8024. DOI: 10.1039/C9SC01928F.

(25) Gao, K.; Nguyen, D. D.; Tu, M.; Wei, G.-W. Generative Network Complex for the Automated Generation of Drug-like Molecules. *J. Chem. Inf. Model.* **2020**, *60*, 5682–5698. DOI: 10.1021/acs.jcim.0c00599. Published Online: Aug. 7, 2020.

(26) Nikolova, N.; Jaworska, J. Approaches to Measure Chemical Similarity - a Review. *QSAR Comb. Sci.* **2003**, *22*, 1006–1026. DOI: 10.1002/qsar.200330831.

(27) Maldonado, A. G.; Doucet, J. P.; Petitjean, M.; Fan, B.-T. Molecular Similarity and Diversity in Chemoinformatics: from Theory to Applications. *Mol. Divers.* **2006**, *10*, 39–79. DOI: 10.1007/s11030-006-8697-1.

(28) Feher, M.; Schmidt, J. M. Property Distributions: Differences between Drugs, Natural Products, and Molecules from Combinatorial Chemistry. *J. Chem. Inf. Comput. Sci.* **2003**, *43*, 218–227. DOI: 10.1021/ci0200467.

(29) Brown, N.; Fiscato, M.; Segler, M. H. S.; Vaucher, A. C. GuacaMol: Benchmarking Models for de Novo Molecular Design. *J. Chem. Inf. Model.* **2019**, *59*, 1096–1108. DOI: 10.1021/acs.jcim.8b00839.

(30) Weininger, D. SMILES, a Chemical Language and Information System. 1. Introduction to Methodology and Encoding Rules. *J. Chem. Inf. Model.* **1988**, *28*, 31–36. DOI: 10.1021/ci00057a005.

(31) Creton, B. Chemoinformatics at IFP Energies Nouvelles:Applications in the Fields of Energy, Transport, and Environment. *Mol. Inf.* **2017**, *36*, 1700028. DOI: 10.1002/minf.201700028.

(32) Weininger, D.; Weininger, A.; Weininger, J. L. SMILES. 2. Algorithm for Generation of Unique SMILES Notation. *J. Chem. Inf. Model.* **1989**, *29*, 97–101. DOI: 10.1021/ci00062a008.

(33) Landrum, G. *RDKit: Open-Source Cheminformatics*. http://www.rdkit.org/ (accessed 2021-04-26).

(34) Ruggiu, F.; Marcou, G.; Varnek, A.; Horvath, D. ISIDA Property-Labelled Fragment Descriptors. *Mol. Inf.* **2010**, *29*, 855–868. DOI: 10.1002/minf.201000099.

(35) Varnek, A.; Fourches, D.; Horvath, D.; Klimchuk, O.; Gaudin, C.; Vayer, P.; Solov'ev, V.; Hoonakker, F.; Tetko, I.; Marcou, G. ISIDA - Platform for Virtual Screening Based on Fragment and Pharmacophoric Descriptors. *Curr. Comput. Aided Des.* **2008**, *4*, 191–198. DOI: 10.2174/157340908785747465.

(36) Muller, C.; Maldonado, A. G.; Varnek, A.; Creton, B. Prediction of Optimal Salinities for Surfactant Formulations Using a Quantitative Structure–Property Relationships Approach. *Energy Fuels* **2015**, *29*, 4281–4288. DOI: 10.1021/acs.energyfuels.5b00825.

(37) Drucker, H.; Burges, C. J. C.; Kaufman, L.; Smola, A.; Vapnik, V. Support Vector Regression Machines. In *Proceedings of the 9th International Conference on Neural Information Processing Systems*; MIT Press: Cambridge, MA, USA, 1996; pp 155–161.

(38) Chang, C.-C.; Lin, C.-J. LIBSVM:A Library for Support Vector Machines. *ACM Transactions on Intelligent Systems and Technology* **2011**, *2*, 1–27.

(39) Hiot, L. M.; Ong, Y. S.; Tenne, Y.; Goh, C.-K. *Computational Intelligence in Expensive Optimization Problems*, Vol. 2; Springer Berlin Heidelberg, 2010. DOI: 10.1007/978-3-642-10701-6.

(40) Zabinsky, Z. B. *Stochastic Adaptive Search for Global Optimization*; Nonconvex optimization and its applications; Kluwer Academic Publishers, 2003.

(41) Langouët, H. Constraints Derivative-Free Optimization:Two Industrial Applications in Reservoir Engineering and in Engine Calibration. Ph.D. Dissertation, Université Nice Sophia Antipolis, 2011. https://tel.archives-ouvertes.fr/tel-00671987 (accessed 2021-04-26).

(42) Sinoquet, D.; Langouët, H.; Da Veiga, S. A Derivative Free Optimization Method for Reservoir Characterization Inverse Problem. In *72nd EAGE Conference and Exhibition incorporating SPE EUROPEC 2010*; European Association of Geoscientists & Engineers, 2010. DOI: 10.3997/2214-4609.201401002.

(43) Gramatica, P. Principles of QSAR Models Validation:Internal and External. *QSAR Comb. Sci.* **2007**, *26*, 694–701. DOI: 10.1002/qsar.200610151.

(44) Lunghini, F.; Marcou, G.; Gantzer, P.; Azam, P.; Horvath, D.; van Miert, E.; Varnek, A. Modelling of Ready Biodegradability Based on Combined Public and Industrial Data Sources. *SAR QSAR Environ. Res.* **2019**, *31*, 171–186. DOI: 10.1080/1062936X.2019.1697360.

(45) Roy, K.; Kar, S.; Ambure, P. On a Simple Approach for Determining Applicability Domain of QSAR Models. *Chemom. Intell. Lab. Syst.* **2015**, *145*, 22–29. DOI: 10.1016/j.chemolab.2015.04.013.

(46) Dragos, H.; Gilles, M.; Alexandre, V. Predicting the Predictability: a Unified Approach to the Applicability Domain Problem of QSAR Models. *J. Chem. Inf. Model.* **2009**, *49*, 1762–1776. DOI: 10.1021/ci9000579.

(47) Wold, S.; Esbensen, K.; Geladi, P. Principal Component Analysis. *Chemom. Intell. Lab. Syst.* **1987**, *2*, 37–52. DOI: 10.1016/0169-7439(87)80084-9.

(48) Fredenslund, A.; Jones, R. L.; Prausnitz, J. M. Group-Contribution Estimation of Activity Coefficients in Nonideal Liquid Mixtures. *AIChE J.* **1975**, *21*, 1086–1099. DOI: 10.1002/aic.690210607.

(49) Derringer, G. C.; Markham, R. L. A Computer-Based Methodology for Matching Polymer Structures with Required Properties. *J. Appl. Polym. Sci.* **1985**, *30*, 4609–4617. DOI: 10.1002/app.1985.070301208.

(50) Pretel, E. J.; López, P. A.; Bottini, S. B.; Brignole, E. A. Computer-Aided Molecular Design of Solvents for Separation Processes. *AIChE J.* **1994**, *40*, 1349–1360. DOI: 10.1002/aic.690400808.

(51) Nilakantan, R.; Bauman, N.; Venkataraghavan, R. A Method for Automatic Generation of Novel Chemical Structures and its Potential Applications to Drug Discovery. *J. Chem. Inf. Model.* **1991**, *31*, 527–530. DOI: 10.1021/ci00004a016.

(52) Kullback, S.; Leibler, R. A. On Information and Sufficiency. *Ann. Math. Statist.* **1951**, *22*, 79–86. DOI: 10.1214/aoms/1177729694.

(53) Bhattacharyya, A. On a Measure of Divergence Between Two Statistical Populations Defined by their Probability Distributions. *Bulletin of the Calcutta Mathematical Society* **1943**, *35*, 99–109.

(54) Shemyakin, A. Hellinger Distance and Non-informative Priors. *Bayesian Anal.* **2014**, *9*, 923–938. DOI: 10.1214/14-BA881.

(55) Saldana, D. A.; Starck, L.; Mougin, P.; Rousseau, B.; Pidol, L.; Jeuland, N.; Creton, B. Flash Point and Cetane Number Predictions for Fuel Compounds Using Quantitative Structure Property Relationship (QSPR) Methods. *Energy Fuels* **2011**, *25*, 3900–3908. DOI: 10.1021/ef200795j.

(56) Vidal, M.; Rogers, W. J.; Holste, J. C.; Mannan, M. S. A Review of Estimation Methods for Flash Points and Flammability Limits. *Proc. Safety Prog.* **2004**, *23*, 47–55. DOI: 10.1002/prs.10004.

(57) Levy, J. M. *Hazmat Chemistry Study Guide (Second Edition)*; Firebelle Productions, 2005.

(58) Carroll, F. A.; Lin, C.-Y.; Quina, F. H. Improved Prediction of Hydrocarbon Flash Points from Boiling Point Data. *Energy Fuels* **2010**, *24*, 4854–4856. DOI: 10.1021/ef1005836.

(59) Catoire, L.; Naudet, V. A Unique Equation to Estimate Flash Points of Selected Pure Liquids Application to the Correction of Probably Erroneous Flash Point Values. *Journal of Physical and Chemical Reference Data* **2004**, *33*, 1083–1111. DOI: 10.1063/1.1835321.

(60) Patel, S. J.; Ng, D.; Mannan, M. S. QSPR Flash Point Prediction of Solvents Using Topological Indices for Application in Computer Aided Molecular Design. *Ind. Eng. Chem. Res.* **2009**, *48*, 7378–7387. DOI: 10.1021/ie9000794.

(61) Patel, S. J.; Ng, D.; Mannan, M. S. QSPR Flash Point Prediction of Solvents Using Topological Indices for Application in Computer Aided Molecular Design. *Ind. Eng. Chem. Res.* **2010**, *49*, 8282–8287. DOI: 10.1021/ie101378h.

(62) Rowley, R. L.; Wilding, W. V.; Oscarson, J. L.; Yang, Y.; Zundel, N. A.; Daubert, T. E.; Danner, R. P. DIPPR Data Compilation of Pure Compound Properties. *Design Institute for Physical Properties* **2003**.

(63) Ertl, P.; Schuffenhauer, A. Estimation of Synthetic Accessibility Score of Drug-Like Molecules Based on Molecular Complexity and Fragment Contributions. *J. Cheminformat.* **2009**, *1*, 8. DOI: 10.1186/1758-2946-1-8.

(64) Inoue, T.; Tanaka, K.; Kotera, M.; Funatsu, K. Improvement of the Structure Generator DAECS with Respect to Structural Diversity. *Mol. Inf.* **2021**, *40*, 2000225. DOI: 10.1002/minf.202000225.

(65) Pearlman, R. S. Novel Software Tools for Chemical Diversity. *Perspec. Drug Discov. Des.* **1998**, *9*, 339–353. DOI: 10.1023/A:1027232610247.

(66) Bishop, C. M.; Svensén, M.; Williams, C. K. I. GTM:The Generative Topographic Mapping. *Neural Comput.* **1998**, *10*, 215–234. DOI: 10.1162/089976698300017953.

(67) Kireeva, N.; Baskin, I. I.; Gaspar, H. A.; Horvath, D.; Marcou, G.; Varnek, A. Generative Topographic Mapping (GTM):Universal Tool for Data Visualization, Structure-Activity Modeling and Dataset Comparison. *Mol. Inf.* **2012**, *31*, 301–312. DOI: 10.1002/minf.201100163.

(68) Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; Vanderplas, J.; Passos, A.; Cournapeau, D.; Brucher, M.; Perrot, M.; Duchesnay, E. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* **2011**, *12*, 2825–2830.

(69) Virtanen, P.; Gommers, R.; Oliphant, T. E.; Haberland, M.; Reddy, T.; Cournapeau, D.; Burovski, E.; Peterson, P.; Weckesser, W.; Bright, J.; van der Walt, S. J.; Brett, M.; Wilson, J.; Millman, K. J.; Mayorov, N.; Nelson, A. R. J.; Jones, E.; Kern, R.; Larson, E.; Carey, C. J.; Polat, VanderPlas, Jake; Laxalde, D.; Perktold, J.; Cimrman, R.; Henriksen, I.; Quintero, E. A.; Harris, C. R.; Archibald, A. M.; Ribeiro, A. H.; Pedregosa, F.; van Mulbregt, P.; SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods* **2020**, *17*, 261–272. DOI: 10.1038/s41592-019-0686-2.

(70) Hunter, J. D. Matplotlib: A 2D Graphics Environment. *Computing in Science & Engineering* **2007**, *9*, 90–95. DOI: 10.1109/MCSE.2007.55.