MASSA Algorithm: an automated rational sampling of training and test subsets for QSAR modeling

Gabriel Corrêa Veríssimo[1], Simone Queiroz Pantaleão[2], Philipe de Oliveira Fernandes[1], Jadson Castro Gertrudes[3], Thales Kronenberger[4], Kathia Maria Honorio[2,5], Vinícius Gonçalves Maltarollo[1]*

1. Department of Pharmaceutical Products, Faculty of Pharmacy, Federal University of Minas Gerais, Belo Horizonte, MG 31270-901, Brazil.
2. Center of Human and Natural Sciences, Federal University of ABC, Santo André, SP 09210-170, Brazil.
3. Department of Computing, Institute of Exact and Biological Sciences, Federal University of Ouro Preto, Ouro Preto, MG 35400-000, Brazil.
4. Department of Internal Medicine VIII, University Hospital Tübingen, Tübingen, Germany.
5. School of Arts, Sciences and Humanities, University of São Paulo, São Paulo, SP 05508-000, Brazil.

*** Corresponding author**: viniciusmaltarollo@gmail.com

ORCID iD

Gabriel Corrêa Veríssimo: https://orcid.org/0000-0001-7480-7198

Simone Queiroz Pantaleão: https://orcid.org/0000-0002-5183-7906

Philipe de Oliveira Fernandes: https://orcid.org/0000-0001-8089-2958

Jadson Castro Gertrudes: https://orcid.org/0000-0002-0861-6681

Thales Kronenberger: https://orcid.org/0000-0001-6933-7590

Kathia Maria Honorio: https://orcid.org/0000-0002-6938-0676

Vinícius Gonçalves Maltarollo: https://orcid.org/0000-0001-9675-5907

DECLARATIONS

Consent to participate

Not applicable.

Consent for publication

Not applicable.

Ethics approval

Not applicable.

Competing interests

The authors have no competing interests (financial or non-financial) to declare that are relevant to the content of this article.

Funding

Conselho Nacional de Desenvolvimento Científico e Tecnológico, Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CNPq), Fundação de Amparo à Pesquisa do Estado de Minas Gerais (FAPEMIG), Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP), Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) and Pró-Reitoria de Pesquisa of the Universidade Federal de Minas Gerais for financial support and academic grants. OpenEye Scientific Software for OMEGA and QUACPAC academic licenses.

Code availability

The source code is available in the first author's GitHub repository at https://github.com/gcverissimo/MASSA_Algorithm.

Data availability

The molecular files are available in the first author's GitHub repository at https://github.com/gcverissimo/MASSA_datasets.

ACKNOWLEDGMENTS

The authors would like to thank Conselho Nacional de Desenvolvimento Científico e Tecnológico, Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CNPq), Fundação de Amparo à Pesquisa do Estado de Minas Gerais (FAPEMIG), Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP), Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) and Pró-Reitoria de Pesquisa of the Universidade Federal de Minas Gerais for financial support, OpenEye Scientific Software for OMEGA and QUACPAC academic licenses and Prof. Dr. Raquel Cardoso de Melo Minardi for her encouragement and for offering the course in which this tool was developed.

ABSTRACT

QSAR models capable of predicting biological, toxicity, and pharmacokinetic properties were widely used to search lead bioactive molecules in chemical databases. The dataset's preparation to build these models has a strong influence on the quality of the generated models, and sampling requires that the original dataset be divided into training (for model training) and test (for statistical evaluation) sets. This sampling can be done randomly or rationally, but the rational division is superior. In this paper, we present MASSA, a Python tool that can be used to automatically sample datasets by exploring the biological, physicochemical, and structural spaces of molecules using PCA, HCA, and K-modes. The proposed algorithm is very useful when the variables used for QSAR are not available or to construct multiple QSAR models with the same training and test sets, producing models with lower variability and better values for validation metrics. These results were obtained even when the descriptors used in the QSAR/QSPR were different from those used in the separation of training and test sets, indicating that this tool can be used to build models for more than one QSAR/QSPR technique. Finally, this tool also generates useful graphical representations that can provide insights into the data.

Keywords: Clustering, Hierarchical clustering analysis (HCA), K-modes, Training and test sampling, QSAR, Computer-aided drug design, Python.

INTRODUCTION

Research and development (R&D) in the context of drug design faces many challenges, such as high cost, limited number of substances that can be tested, difficulty in finding new active chemical structures, limited success in preclinical and clinical trials, and long periods to reach the market [1]. One solution to these problems is computer-aided drug design (CADD), which is constantly evolving in terms of new techniques, algorithms, and software. CADD has become an essential part of the drug planning process by speeding up, reducing costs, and increasing the success rate [2]. The quantitative structure-activity relationship (QSAR) technique is one of CADD methods used in drug development and has gotten a lot of attention because of its applicability in the prediction of pharmacokinetic characteristics, biological activity, toxicity, risk assessment, screening of bioactive molecules in chemical databases and lead optimization [2, 3]. Several advances in medicinal chemistry and toxicology have been made with the application of QSAR models, highlighting the development of norfloxacin, the first fluoroquinolone with antibacterial activity to reach the market. In addition, the International Council for Harmonisation of Technical Requirements for Pharmaceuticals for Human Use (ICH) has recommended the use of QSAR models in mutagenicity studies of drug candidates and their degradation products, providing more information and speeding up toxicity studies [4, 5].

The main objective of QSAR studies is to create predictive models built on a mathematical-statistical correlation between structural features (independent variables or descriptors) and the desired property (dependent variable), generally a biological activity, however, other types of properties, such as physicochemical properties can also be predicted (in this sense, the so-called quantitative structure-property relationship, QSPR) [2, 3, 6]. In fact, QSAR modeling is widely utilized across academic, industrial, and governmental sectors for assessing the potential effects of chemicals, materials, and nanomaterials on human health and ecological systems. It can be used to predict various different properties, such as negative logarithm of the dissociation constant (pK_a), *n*-octanol–water partition coefficients (log P), *ab initio* properties (e. g., dipole moment), reactivity indicators (e.g., E_{HOMO} e E_{LUMO}), *in vitro* toxicity, multi-target *in vitro* toxicity, *in vivo* rat oral toxicity, drug metabolizing via cytochrome P450, reactivity (reaction rate), inhibition of G-protein-coupled receptors, antiviral and antibacterial activities and many others [6–8]. Furthermore, the obtained correlations could be employed to provide mechanistic interpretation for the modeled desired property and guide structural modifications to design more potent compounds. QSAR protocols are based on three fundamental steps: data preparation, model construction, and model validation. The data preparation strongly influences the other ones; in this step, one procedure requires the original dataset must be sampled into training set (used for model training) and test set (used to further evaluate the predictive ability of the model) [3]. This sampling can be done randomly or rationally, but the rational or purposeful division is superior to the random method [2, 6, 9–11].

Sampling steps in any analysis are critical for data quality and, consequently, for the quality of information derived from these data, since overall quality is closely related to the data representativeness [10]. Although random sampling can produce predictive models and its overall predictability is often comparable to rational selection, validation metrics show discrepancies when obtained by models built with the same randomly sampled dataset. This suggests that these models have low reproducibility and can lead to erroneous conclusions and only representative samples can provide reliable analytical information because the values may be too optimistic or too pessimistic, depending on the representativeness of this separation [3, 6, 10].

Compared to random selection, rational sampling of compounds with chemical properties covering the entire population of available data does not introduce any bias. It provides models that are more accurate and have a wider applicability domain [11]. The rational selection of compounds leads to the development of models that produce more reliable results and better values in internal and external validation metrics, allowing precise and statistically significant structure-activity correlations that would otherwise be insignificant due to random selection [3, 6, 9, 12, 13].

In QSAR context, many algorithms are used for the rational division of the dataset into training and test sets, including K-means clustering, based on activity selection, D-optimal design, hierarchical clustering analysis (HCA), Kennard-Stone algorithm, minimal test set dissimilarity, Self-Organizing Maps (SOM, also known as Kohonen maps), and sphere exclusion algorithm [2, 3, 6, 14–17]. All of them are based on two principles: the training set must be structurally diverse enough to cover the entire descriptor space of the overall dataset, and the compounds in the training and test sets must be close together [6]. In general, among the algorithms used for rational sampling, there is no statistically significant difference, i.e., the effect on the statistical performance of a QSAR model is small. However, the predictive power of a QSAR model is strongly determined by the distribution of the training compounds in the chemical space, and ideally, the compounds should be uniformly distributed throughout the space [2, 9]. Furthermore, different QSAR methodologies utilize different descriptors, and the choice and use of these descriptors in sampling can bias the models. This can be observed in studies that use more than one QSAR technique, as well as when a screening of descriptors is performed to determine which descriptor can generate the best models.

In the literature, several works can be found describing the construction of predictive models for the most diverse biological activities using different QSAR techniques, including Hologram-QSAR (HQSAR), Comparative Molecular Field Analysis (CoMFA), Comparative Molecular Similarity Indices Analysis (CoMSIA), LQTA-QSAR (LQTA, Laboratório de Quimiometria Teórica e Aplicada), Multitarget-based-QSAR (e.g., QSAR-Co and QSAR-Co-X) and Machine Learning-Based-QSAR employing several algorithms, such as Random Forest (RF), Support Vector Machine (SVM), Gradient Boosting (GB), and Artificial Neural Networks (ANN) [15–27]. Several of these studies apply rational sampling in training and test sets based on hierarchical clustering analysis (HCA) of biological, physicochemical and structural chemical spaces. Although there are several implementations of these different algorithms in the literature, little effort has been put into developing open-source tools that automatically execute one of these algorithms to perform the splitting of the compound set into training and test sets based on the entire chemical space (including both dependent and independent variables of the dataset).

We present in this paper an open-source, easy-to-use Python tool called MASSA Algorithm ("Molecular dAta Set SAmpling Algorithm") [28] to perform the automatic sampling of datasets of molecules into training and test sets. This algorithm is based on hierarchical clustering analysis of physicochemical and structural spaces, as well as the dependent variables (biological activities). This tool, powered by RDKit [29], only needs a ".sdf" file with molecules and their respective structures and biological activities to select them proportionally to their physicochemical, structural, and biological diversity. MASSA uses the biological activity of compounds together with calculated Atom Pairs fingerprint and physicochemical properties, such as the number of hydrogen bond acceptors (HBA), the number of hydrogen bond donors (HBD), molar weight (MW), the number of rotatable bonds (nRotB), sp^3 carbon fraction (Fsp^3), topological polar surface area (TPSA) and Wildman-Crippen partition coefficient (log P). The algorithm employs the best-practice procedures for dataset sampling improving the overall quality of models build with any QSAR technique and generating graphical view of the distribution of these molecules. In addition to automatically generated plots (that can be useful for discussing the results obtained from the construction and interpretation of models), this tool allows the advanced user to explore and change algorithm parameters without sacrificing the basic user experience, which will use the default settings. We also presented a comparative analysis with multiple datasets between the results obtained from QSAR randomly sampled models and QSAR rationally sampled models using the proposed algorithm to introduce, describe, and validate the utility, reliability, and application of MASSA.

MATERIALS AND METHODS

Implementation

System Overview

MASSA Algorithm is a Python script that calculates physicochemical (HBA, HBD, MW, nRotB, Fsp3, TPSA, and log P) and structural (Atom Pairs fingerprint) properties for a set of molecules contained in an SDF file. By using HCA, each one of these two domains (molecular properties), as well as the domain associated with biological activities, are individually and automatically clustered. A new clustering is performed with the previous cluster labels of the three domains using k-modes [30], resulting in a single cluster division that will be utilized to guide the rational sampling of the molecules. An SDF file containing the computed properties and the distribution of molecules is generated at the end of the sampling procedure together with graphical representations of the distribution and frequency of molecules in the clusters and subsets. The methodology used in each stage of the process is described below and summarized in **Figure 1**.

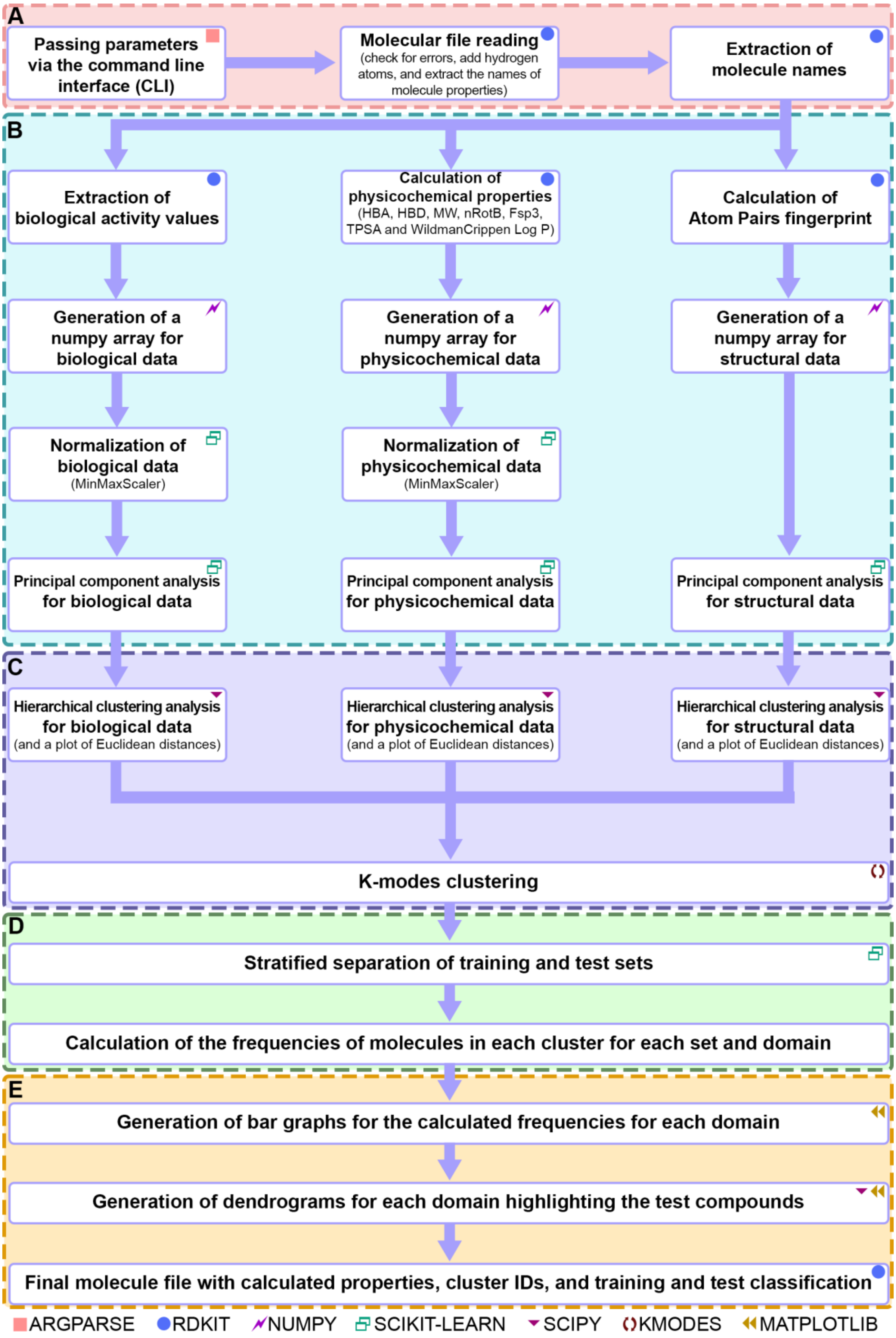
A
Passing parameters via the command line interface (CLI)
Molecular file reading (check for errors, add hydrogen atoms, and extract the names of molecule properties)
Extraction of molecule names
B
Extraction of biological activity values
Calculation of physicochemical properties (HBA, HBD, MW, nRotB, Fsp3, TPSA and WildmanCrippen Log P)
Calculation of Atom Pairs fingerprint
Generation of a numpy array for biological data
Generation of a numpy array for physicochemical data
Generation of a numpy array for structural data
Normalization of biological data (MinMaxScaler)
Normalization of physicochemical data (MinMaxScaler)
Principal component analysis for biological data
Principal component analysis for physicochemical data
Principal component analysis for structural data
C
Hierarchical clustering analysis for biological data (and a plot of Euclidean distances)
Hierarchical clustering analysis for physicochemical data (and a plot of Euclidean distances)
Hierarchical clustering analysis for structural data (and a plot of Euclidean distances)
K-modes clustering
D
Stratified separation of training and test sets
Calculation of the frequencies of molecules in each cluster for each set and domain
E
Generation of bar graphs for the calculated frequencies for each domain
Generation of dendrograms for each domain highlighting the test compounds
Final molecule file with calculated properties, cluster IDs, and training and test classification
ARGPARSE
RDKIT
NUMPY
SCIKIT-LEARN
SCIPY
KMODES
MATPLOTLIB

Figure 1. Illustration of the overall strategy used in the MASSA Algorithm for sampling the entire chemical and biological space of molecules. The symbols labeled in the image represent the main modules used in the Python script for each step. The pandas module was also used to organize the data in multiple steps. A) Initial reading of the molecules file. B) Carrying out pre-processing before clustering. Data extraction for biological activity, physicochemical and fingerprint calculations, data normalization, and dimensionality reduction. C) Individual clustering of each domain (biological, structural, and physicochemical) and the final clustering. D) Stratified division of molecules between training and test sets, followed by a calculation of distribution frequencies between sets, domains, and clusters. E) Generation of plots and the final molecule file.

Command line interface

The first step involved in the construction of the algorithm is related to its interaction with the user to get the information needed in the process. For this, the argparse module [31] was used, which allows the development of easy-to-use command line interfaces and also generates useful help messages.

The following information is obtained from argparse: path to the input and output files, percentage of molecules in each subset, number of biological activities and which columns they represent, the image file format of the generated representations, x-axis font size of the generated diagrams, number of principal components used in Principal Component Analysis (PCA), Singular Value Decomposition (SVD) solver parameter, and linkage method used in HCA. The input and output files are required parameters and the others can be declared or not, since the script can be run with default values, making it easier for basic users without compromising the experience of more advanced users. The number of principal components in the default settings is automatically determined by analysis of variance, as discussed further below, and the default font sizes were determined by multiple visual tests. According to prior studies, the HCA complete linkage method, the percentages of molecules in the training and test sets, and other parameters are used as the default.

To achieve the goal of providing an easy-to-use interface for basic users, if the user does not provide the number and name of biological activities in the SD file, the script will behave as follows: the number of biological activities will be one, and the script will examine how many properties are described in the input file. Except for the name of the molecules, if the number of biological activities is equal to the number of described properties, these properties are considered biological activities automatically; otherwise, an input() function prompts the user to enter the name of the biological activities. The roles of the other parameters are discussed in detail in their respective sections.

Reading SD files

The input file in an SD format is read by RDKit calling the AllChem.SDMolSupplier() function. Then, each molecule is instantiated in a for loop to add the hydrogen atoms, preserving the 3D coordinates of the original file using of AllChem.AddHs() and passes True as the value of the addCords function argument. If an error occurs while reading the molecules, the script closes and the error is reported on the command line and in a log file; otherwise, the data processing and information extraction from the SD file will continue.

The data is organized using pandas.DataFrame object and each row are identified by the name of the molecule extracted by an RDKit function called GetProp(), which is also used to extract the values of the biological activities selected in subsequent steps.

Descriptors' calculation and *pre-processing for clustering*

Since the name and biological activities of the molecules were extracted from the SD file, data on the physicochemical and structural spaces are still lacking to sample the entire chemical space. Fortunately, RDKit has functions to calculate certain fingerprints and descriptors that can be used to represent the structural and physicochemical domains, respectively. Therefore, Atom Pairs fingerprint was calculated using functions provided by rdMolDescriptors and cDataStructs modules, and HBA, HBD, MW, nRotB, Fsp3, TPSA, and WildmanCrippen Log P were calculated using the functions of the module “Descriptors”.

Once the variables related to the biological, physicochemical, and structural spaces were calculated and organized in the data frame, it is possible to start organizing these data and then performing the clustering. An numpy.array was generated for each one of the three domains and the values of each variable, except the fingerprint, were normalized with the MinMaxScaler() function from the scikit-learn package [32].

In the next step, PCA was performed for the arrays that present the number of properties greater than the number of main components desired. For this, the scikit-learn package by the PCA() function was also used. In the step of passing command line arguments, the user can select the number of principal components and the svd_solver argument for the PCA() function. The default value for the svd_solver argument is ‘full’, which implies that the exact full SVD is computed using the standard LAPACK solver, and the components are selected by post-processing [33]. With svd_solver = ‘full’, the number of user-defined principal components is interpreted as follows: if the user-defined number is a decimal number between 0 and 1, the number of components is chosen so that the amount of variance to be explained is greater than the specified percentage;

otherwise, the number of components is the exact integer [33]. The number of principal components is 0.85 in the default settings, which means that the number of principal components is chosen to explain more than 85% of the variance of the dataset.

Clustering

After the data has been pre-processed, HCA can now be performed. First, hierarchical clustering analyses are separately performed on each of the three matrices (biological activities, fingerprints, and physicochemical properties). The linkage matrix for each HCA is built using the linkage() function from the cluster.hierarchy module of the scipy library [34]. The complete linkage method is the default, but the user can choose other options, such as complete, single, ward, average, weighted, centroid, and median.

Euclidean distances are then calculated using the maxdists() function from scipy.cluster.hierarchy that calculates the maximum distances between any cluster [35]. These distances are plotted in a scatter graph and used in the elbow method to calculate the optimal clustering cutoff distance and the number of clusters. The elbow method is implemented by calculating the distance between each data point and a straight line drawn between the first point (representing a single cluster) and the twenty-first point (representing a division into 21 clusters), as described and further explained in **Equation 1**. The points on the x-axis represent the number of clusters, while the points on the y-axis represent the maximum Euclidean distance among the clusters. When the increase in the number of clusters (x) does not significantly affect the difference in Euclidean distance (y), a curve in the shape of an elbow is formed. This elbow-shaped curve indicates the optimal cutoff distance and the minimum number of clusters.

$$\text{distance} = \frac{|(y_2 - y_1)x_0 - (x_2 - x_1)y_0 + x_2y_1 - x_1y_2|}{\sqrt{(y_2 - y_1)^2 + (x_2 - x_1)^2}} \qquad (\textbf{Equation 1})$$

In **Equation 1**, the coordinates x (number of clusters) and y (Euclidean distance) of the first point on the line are x_1 and y_1, respectively, while x_2 and y_2 correspond to the coordinates of the final one. Each one of the coordinates x_0 and y_0 will be used to compute the distance of each observation to the defined line. The curve point farthest from the straight line is then identified as the elbow, indicating the appropriate number of clusters and the correct cutoff distance for HCA. With these values, the flat clusters from the hierarchical clustering are generated by the fcluster() function from scipy.cluster.hierarchy [36].

The dendrograms for each one of these analyses are only created after sampling in training and test to allow the identification in the representation of molecules that belong to the test. The dendrogram() function, which is also part of scipy.cluster.hierarchy, was used for this, and several graphical changes were made with the help of the matplotlib library to improve the presentation of the dendrogram and to generate relevant legends and identifications [37].

A new clustering utilizing k-modes was then done to gather the classification data from the three analyses in a single classification. While k-means algorithm groups continuous numerical variables based on their Euclidean distance, k-modes clusters data points based on the number of matching categories between these points, making it more appropriate for categorical variables. [30]. Thus, the HCA classifications were converted into categorical variables for the k-modes by transforming the integer into a string and appending the term “cluster” at the beginning of each string.

The k-modes clustering is then performed with the number of clusters (x) ranging from 2 to 20, and the clustering cost (y), defined as the sum distance of all points to their respective cluster centroids. This is performed to define the ideal number of clusters by applying the elbow rule implemented with the previously described line-to-point distance equation [30]. After determining the number of clusters, the final cluster division is carried out, and each molecule is assigned to its final k-modes cluster, which will be utilized for sampling in training and test.

Sampling

The selection of training and test sets was performed using the train_test_split() function of the sklearn.model_selection module. This function allows stratified sampling according to the final cluster distribution, ensuring that both sets have similar proportions [38]. The user can specify the percentage of molecules in each set using a two-digit decimal value on the command line. Otherwise, the default values of 80 percent (0.80) for the training set and 20 percent (0.20) for the test set will be used. The frequencies of each cluster in each set (training, test, and total) for the three domains (biological, physicochemical, and structural) and the overall classification are calculated after sampling into training and test sets. The computed frequencies for each one of these four classifications are used to build a bar graph that visually displays the distribution of molecules in clusters and sets.

Finally, a new SDF file is created with RDKit, preserving the previously existing properties, adding the set identification (training or test), the assigned cluster identifications, and the calculated physicochemical properties.

Availability

MASSA Algorithm is fully free under the GNU Affero General Public License (v3) and is accessible in the Python repository: https://pypi.org/project/MASSA-Algorithm/. The source code is also available at the following link: https://github.com/gcverissimo/MASSA_Algorithm.

Before installing this package, it is essential to have Python version 3.8 or higher installed. Additionally, other required packages, such as the RDKit distribution for Python, will be automatically installed during the installation process of this tool. To easily install the MASSA Algorithm, execute the following command: pip install MASSA_Algorithm. Once installed, run the program from the command line with the following command: "MASSA_Algorithm -i <path to the input file> -o <path to the output file>". Alternatively, you can set the optional parameters as well. The complete description of optional parameters can be found in the program's help (command "MASSA_Algorithm -h") or on the source code's home page on GitHub.

Datasets

To analyze the influence of the dataset separation method on the overall quality of the QSAR models, seven datasets of different sizes and nature of dependent variables (biological activities, enzymatic inhibition, and physicochemical properties) were collected from the literature [39–43]. These sets of molecules were chosen by searching for datasets in which HQSAR models had previously been reported, as it is a commercial QSAR technique that is simple to implement and has good reproducibility.

From the seven selected datasets, the following eight activities/properties were used as dependent variables (Y) in the separation and construction of the QSAR models: half maximal inhibitory concentration (IC_{50}) converted to the pIC_{50} scale (- log IC_{50}) against angiotensin-converting enzyme (ACE), dihydrofolate reductase (DHFR), glycogen phosphorylase b (GPB) [39], mantle cell lymphoma (JEKO-1) [40], thyroid hormone receptors TRα and TRβ (TR) [41], cannabinoid receptor subtype CB_1 (a member of G-protein-coupled receptor superfamily – GPCR) [42], and critical micelle concentration (CMC) converted to the pCMC scale (- log CMC) [43]. Except for activities against thyroid hormone receptors TRα and TRβ, in which both subtypes corresponded to the same set of molecules, all activities corresponded to a single separate dataset. **Table 1** lists the properties and the number of molecules in each dataset employed in the analysis.

Table 1. Datasets, targets, dependent variables (y), and the number of molecules (n).

Dataset		Type of target	y	n
ACE [39]		Enzyme	pIC_{50}	114
DHFR [39]		Enzyme	pIC_{50}	361
GPB [39]		Enzyme	pIC_{50}	66
JEKO-1 [40]		Cell	pIC_{50}	35
TR [41]	TRα TRβ	Nuclear receptors	pIC_{50}	68
CB_1 [42]		GPCR receptor	pIC_{50}	75
CMC [43]		Physicochemical Property	pCMC	120

The three-dimensional structures of all compounds in the ACE, DHFR, and GPB datasets were obtained from the original publication's supplementary material; the three-dimensional structures of the molecules in the other datasets were represented in Discovery Studio Visualizer v21.1.0.20298 [44], followed by optimization and correction of the protonation state in biological pH using OMEGA 2.5.1.4 [45, 46] and QUACPAC 1.6.3.1 [47].

Training and test separation

Five types of training-test splits were performed in five replicates for each dataset: a random distribution, a separation using the MASSA Algorithm, and three other rational samplings with Kennard-Stone, Sample Set Partitioning based on Joint X-Y Distances (SPXY) and Sphere Exclusion using astartes [48]. In the distribution with MASSA Algorithm, the following standard parameters were used: percentage of molecules in the training set = 80%, number of biological activities = 1, linkage method = complete, svd_solver = full, and number of principal components = 0.85 (85% of the variance is explained). However, there were a few exceptions: in the TR dataset, the two biological activities (pIC_{50} against TRα and TRβ) were used in the separation and, for the JEKO-1 dataset, the variance to be explained by PCA was 80% because the number of molecules was too small for cluster distribution. Other distributions also considered the proportions of 80% of the molecules for the training set and 20% for the test set. The distance metric used by the astartes distributions was Euclidean distance, and in the Kennard-Stone and SPXY distributions, only one replicate was kept because the result is always the same across replicates. The distance cutoff used for Sphere Exclusion was the default (0.25). All rational sampling techniques are made using Atom Pairs fingerprint to maintain comparability with MASSA sampling. Each dataset's original distribution was also used in the development of models and statistical analyses.

HQSAR models

Models obtained from hologram quantitative structure-activity relationship (HQSAR) were created and validated using Sybyl X 2.1.1 [49] for all data distributions from 2D molecular holograms (independent variable – x) defined by the following parameters: (i) fragments distinguished by considering atoms (A), bonds (B), connections (C), hydrogen atoms (HA), chirality (Ch), and/or hydrogen bond donor/acceptor (DA); (ii) fragment size (F_{size}) and; (iii) hologram length (HL) measured in bins. The parameter combination used for each activity was the one deemed best for that dataset in its reference article to compare the quality of the models generated between the different distributions (**Table 2**). All HQSAR models were created with the maximum number of principal components (PCs) equal to or greater than six, as specified in the reference article, and validated using leave-one-out internal cross-validation (q^2_{LOO}) and other external validation metrics, as described in the **Evaluated metrics** section.

Table 2. HQSAR parameters of each dataset.

Dataset	F_{dist}	F_{size}	HL	PC
ACE [39]	A/B/C/Ch/HA	4 to 7	4999	4
DHFR [39]	A/B/C/Ch/HA	4 to 7	4999	6
GPB [39]	A/B/C	4 to 7	4999	2
JEKO-1 [40]	A/B/Ch/HA	4 to 7	53	6
THR [41] TRα	A/B/C/DA	5 to 8	401	6
TRβ	A/B/C/DA	5 to 8	353	5
CB_1 [42]	A/B/C/D/Ch	4 to 7	71	6
CMC [43]	A/B/C/DA	7 a 10	307	6

F_{dist}: fragment distinction, F_{size}: fragment size, HL: hologram length, PC: number of principal components.

The training and test distributions for all replicates were highlighted in a similarity map analysis based on the coordinates of the HQSAR fragment counts to assess the applicability domain of HQSAR. The coordinates of these maps were also used for HCA employing the complete linkage method and evaluating the concordance between the clusters obtained from the HQSAR fragment counts and those obtained from the various distributions in training and test sets. For this, the percentage of discordance was calculated as follows: the number of total clusters empty at least once in each set (training or test) was divided by the total number of clusters; the result of this division was multiplied by 100, and the percentage of discordance was obtained. The percentage of discordance was then subtracted from 100 to calculate the percentage of concordance.

Random Forest-based-QSAR (RF-QSAR) models

The RF-QSAR models were generated by employing a workflow implemented in the KNIME platform [50] using Atom Pairs fingerprint, 1024 bits, minimum and maximum path lengths equal to 1 and 10, respectively, calculated with RDkit as descriptors for random forest models. The RF-parameters as maximum levels number technique (MaxLevels) and models' number (Nmodels) were set as 10 and 10,000, respectively. All RF-QSAR models were validated with 10-fold internal cross-validation ($q^2_{10\text{-fold}}$) and external validations. The calculated external validations metrics are described in **Evaluated metrics** section.

The applicability domain was assessed by the bounding box approach employing a principal component analysis (PCA) as described in previous work [51] using Scikit-learn [52] and Scipy [53] libraries Atom Pairs fingerprint. The independent variables from the training data were used to build a PCA model and then, applied to transform the test data. The transformed data from training and test sets were applied to perform an applicability domain based on the distances of each test sample from the training. This was done by measuring the Euclidean, Manhattan, Cosine, and Wasserstein (probability distribution) distances, with a threshold of 95%. To reach a consensus among all distance metrics, a sample was designated as 'outside of applicability domain' if it was classified as such by most of the distance metrics (majority of votes). The results were presented as the percentage of samples within the applicability domain in the test data. This procedure was executed for each training/test splits for all datasets.

Evaluated metrics

External validation implies estimating the biological activity of test compounds along with the calculation of quality metrics for regression models. So, we computed the following metrics: Q^2_{F1}, Q^2_{F2}, and Q^2_{F3}, used to measure the model predictability, and root mean square error (RMSE) of the external set, used to measure the model accuracy (a comprehensive review and a detailed description of these metrics can be found in Refs. [54–57]). We also used the Concordance Correlation Coefficient (CCC) to evaluate the relationship between precision (fitting of observed data with fitting line) and accuracy (how deviated the regression line is) [54, 58]. These external validation metrics, and the internal validation metric (q^2_{LOO} for HQSAR and $q^2_{10\text{-fold}}$ for RF-QSAR), were used to build violin plots and to analyze the results.

RESULTS AND DISCUSSION

Output examples

MASSA generates the following visualizations for each distribution of the dataset: three dendrograms related to HCA using biological, structural, and physicochemical properties; three graphs taking into account the Euclidean distances between the clusters previously obtained; four bar graphs (biological, structural, physicochemical, and general spaces) for the percentages of information at each cluster; and a log file with the values of the percentages of each cluster in the domains analyzed. The graphs produced from MASSA for the data distribution on the CMC set are shown in **Figures 2–3**. The colors used to represent the clusters in the dendrogram were validated by the Microsoft Windows color filter to ensure cluster differentiation even for people with different types of color blindness (deuteranopia, protanopia, and tritanopia).

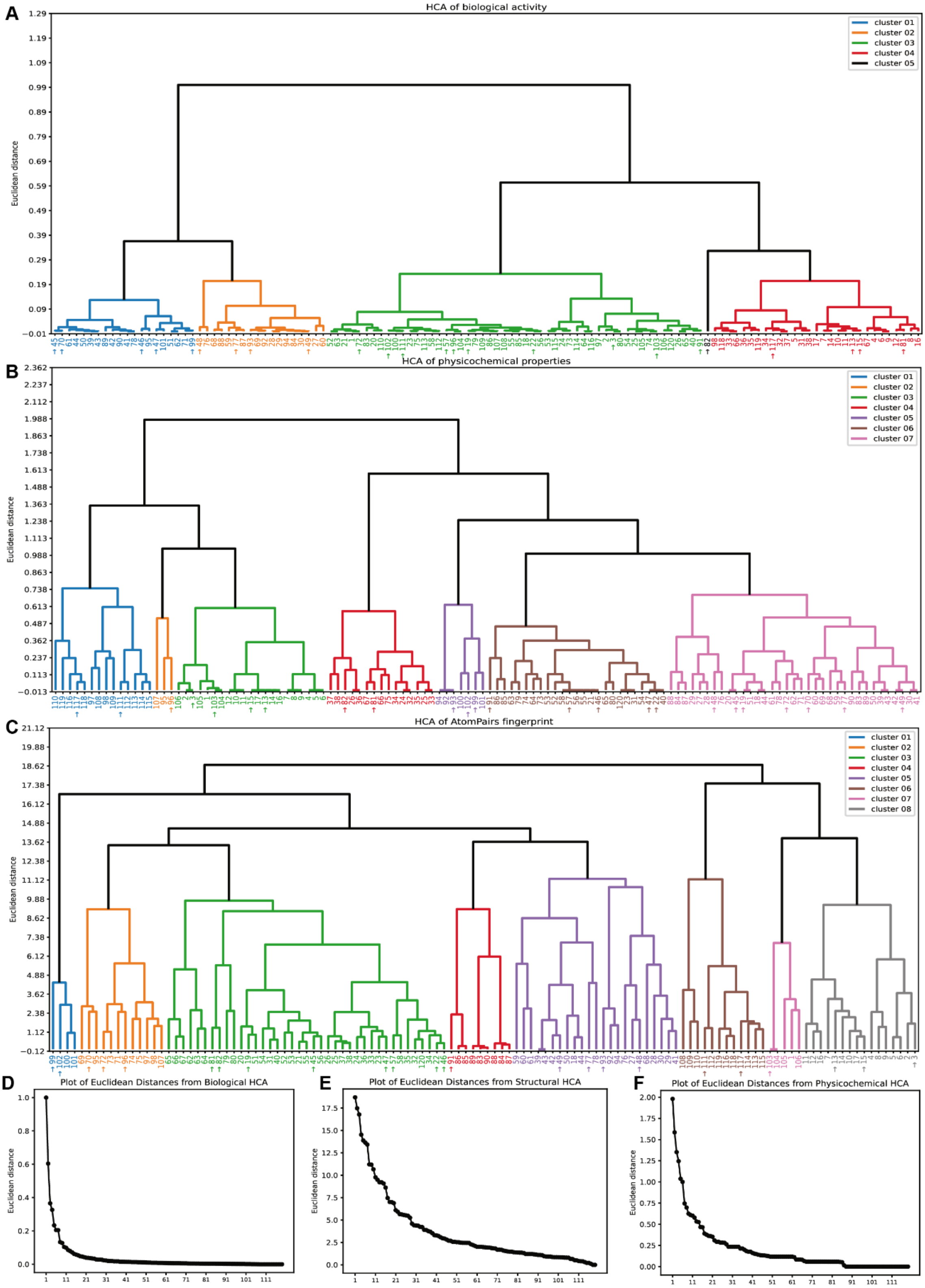

Figure 2. MASSA clustering plots for the CMC set. From A to C, we represent the dendrograms of each domain (biological activity, physicochemical properties, and structural - AtomPairs fingerprint - respectively); from D to F, the line graphs with markers for the Euclidean distances in each domain are represented. In dendrograms, each color represents a different cluster, and arrows on the labels next to the names of the

molecules indicate that the molecule belongs to the test set, while those without arrows belong to the training set.

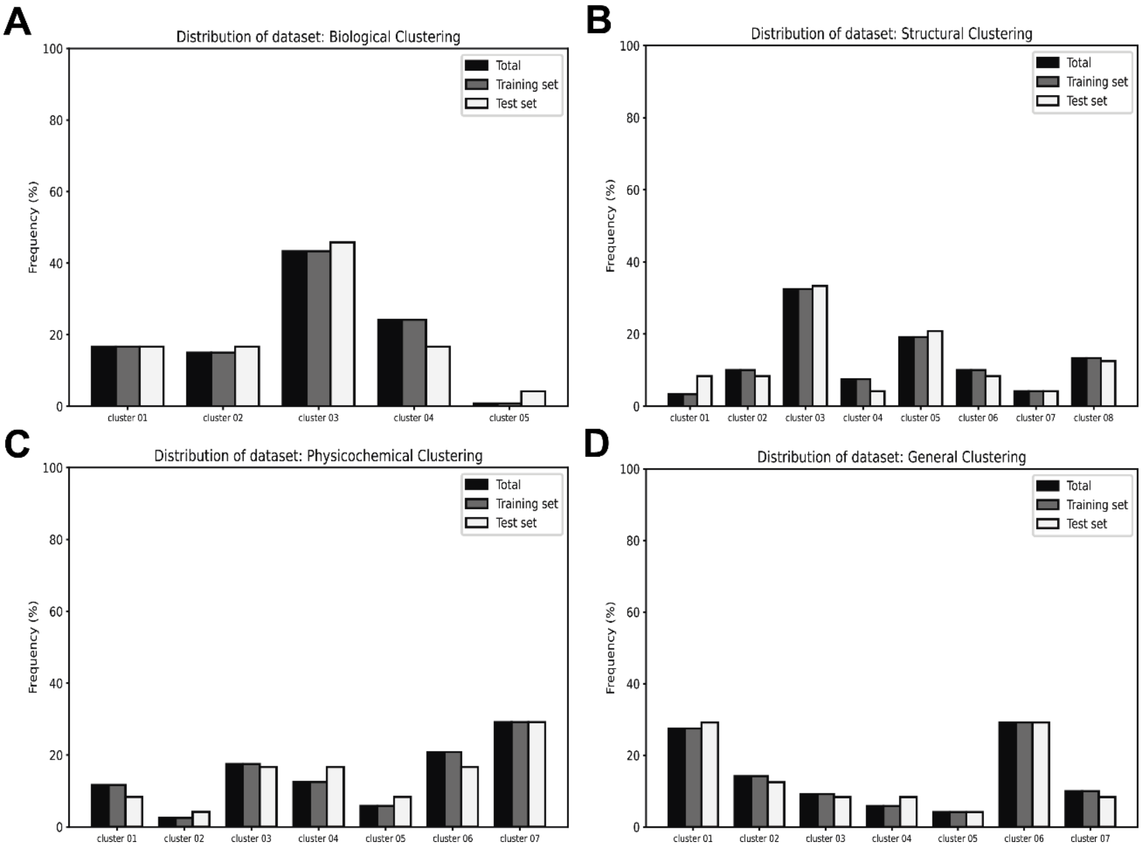

Figure 3. A MASSA-generated bar plot showing the frequency of molecules in each cluster for biological (A), structural (B), physicochemical (C), and general domains (D) of the CMC dataset.

Each image in **Figures 2–3** was created individually and saved as an editable ".svg" vector file. **Figure 2** (**A-C**) dendrograms show that for this dataset, the biological domain had 5 clusters defined, the physicochemical domain had 7, and the structural domain had 8. **Figure 2** (**D-F**) shows the graphs of the Euclidean distance between clusters, proving that the algorithm successfully solves the optimal number of clusters using the elbow rule for this dataset. This demonstrates the achievement of the goal to identify the minimum number of clusters required for explaining the variation in the data. Increasing the number of clusters does not significantly improve the data modeling, which highlights the accuracy and automatic determination of the optimal number of clusters by the proposed algorithm. Moreover, the algorithm effectively handles the distribution in both the training and test sets, maintaining balanced frequencies of molecules across all domains in the training, test, and complete sets, as exemplified in **Figure 3**.

Regarding the other datasets, the algorithm correctly found the optimal number of clusters in all tested cases. Reducing the number of clusters to an acceptable minimum capable of explaining the variation is also necessary to avoid single-molecule clusters, which would make the distribution in training and test sets impossible without loss of representativeness. This is a special problem in datasets with an extremely low number of molecules and high chemical and biological diversity. In these cases, the stratified distribution process becomes complicated, and maintaining frequency correlation between training and test sets becomes extremely difficult. This point is illustrated in **Figure 4**.

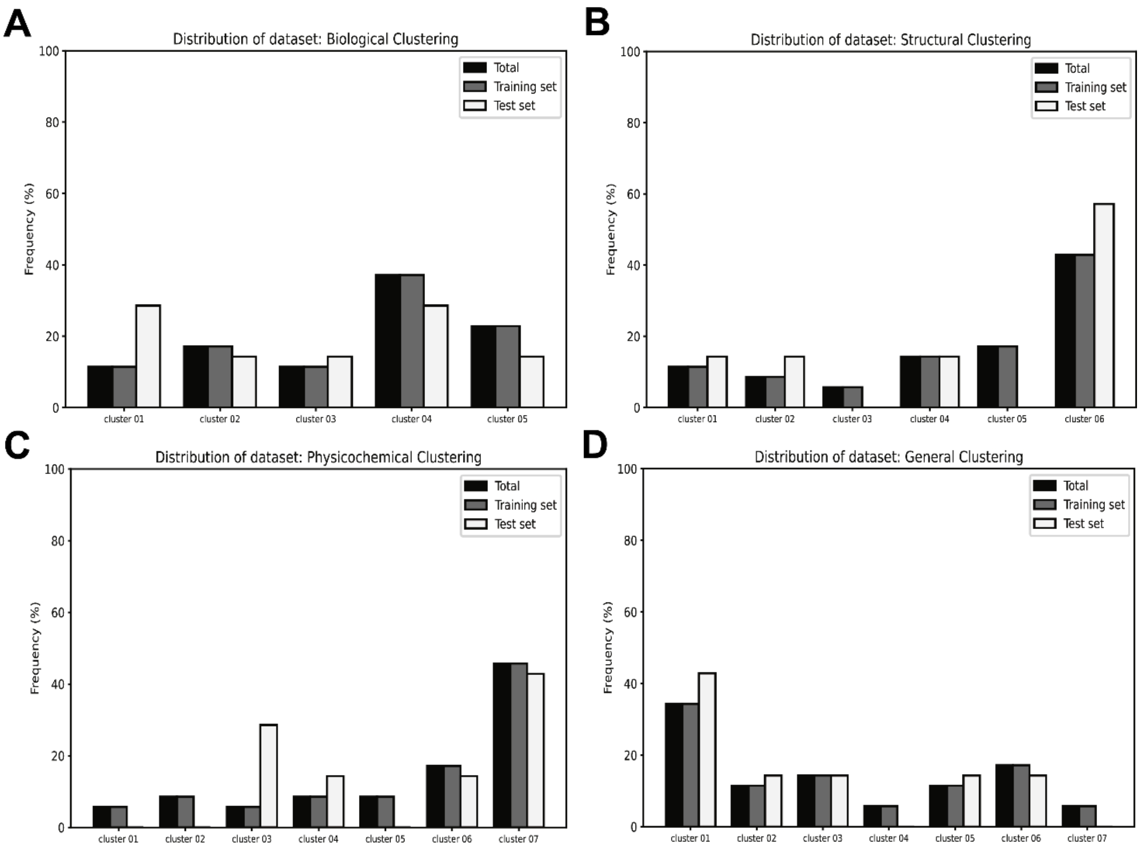

Figure 4. A MASSA-generated bar plot showing the frequency of molecules in each cluster for biological (A), structural (B), physicochemical (C), and general domains (D) of the JEKO-1 dataset.

In this context, the distribution frequencies of the dataset relative to the activity against JEKO-1 cells (dataset with 35 molecules) showed that when the number of molecules is exceptionally very low, the separation process becomes unfeasible while preserving representativeness (see **Figure 4**). It is important to note, although the amount of variation to be explained in PCA can be reduced to obtain a good correlation in the distribution frequency, the low number of molecules in this dataset makes it very hard for constructing predictive QSAR models; this will be covered in more details during the discussion on validation metrics.

Finally, **Figure 5** shows that even for datasets with few molecules (68 molecules) and two biological activities considered in HCA, such as the dataset for thyroid hormone receptors, the frequency of clusters was maintained between the original set and the training and test subsets.

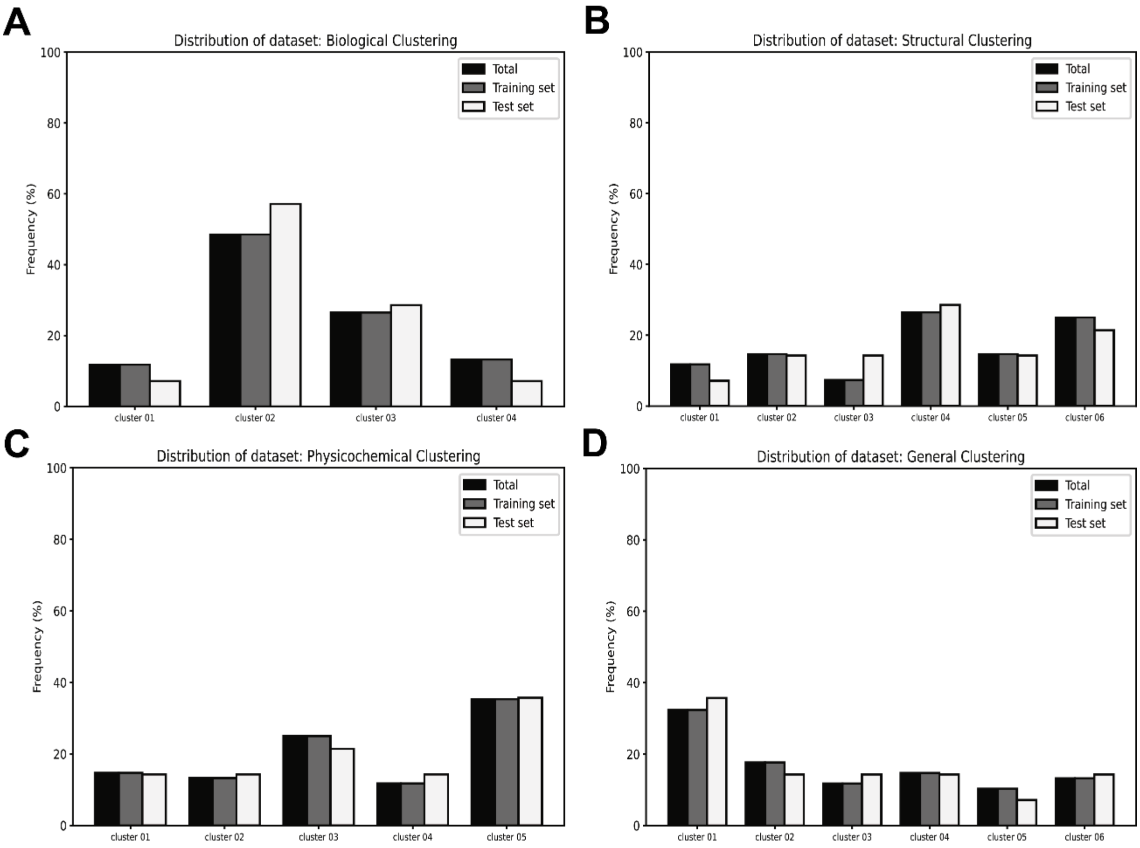

Figure 5. A MASSA-generated bar plot showing the frequency of molecules in each cluster for biological (A), structural (B), physicochemical (C), and general domains (D) of the TR dataset.

Evaluation of applicability domain of QSAR models

The first analysis performed for the HQSAR models was the evaluation of the distribution of samples in similarity maps calculated with the HQSAR fragment counts between training and test sets. The seven datasets and their training-test distributions from MASSA, Kennard-Stone, SPXY, Sphere Exclusion, random, and referential (from the original study) approaches were represented by 126 similarity maps (available in Supplementary Material). These maps demonstrated that sampling using MASSA Algorithm, whose structural features are based on the Atom Pairs fingerprint, was capable of ensuring the representativeness of HQSAR fragment counts better than random, Kennard-Stone and SPXY sampling, and in some cases, even the previous/original work.

Despite the direct observation of the similarity map and the presence of molecules with a proper distribution between map clusters in both sets (training and test), a more reliable analysis was required to evaluate the agreement and the population of these clusters between the sampled sets. For this, map coordinates were used for HCA employing the full linkage method, and the percentages of concordance between HQSAR clusters were calculated and plotted on a violin chart for each distribution (**Figure 6**).

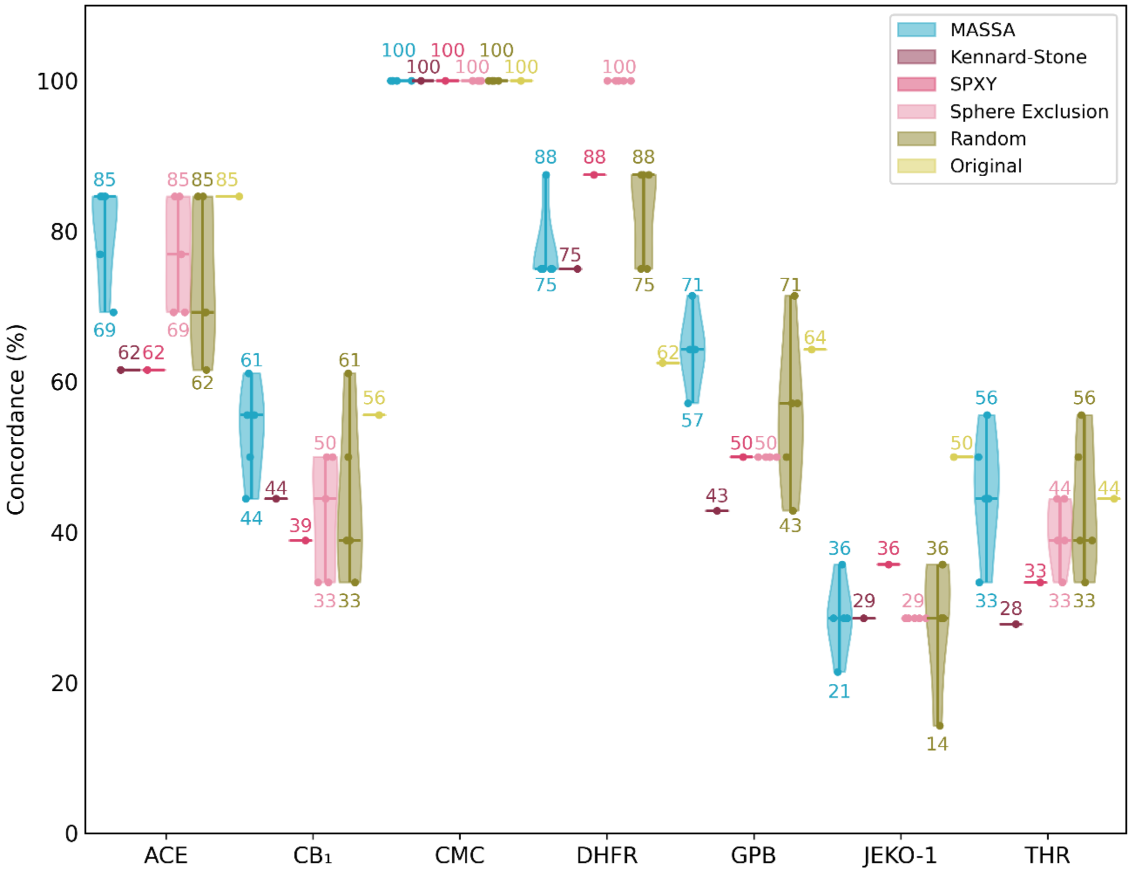

Figure 6. Distribution of the percentage of concordance for the HQSAR descriptor clusters across the training and test sets in different datasets. The text values represent the lowest and highest concordance percentages, while the median is represented by a line of the respective sampling algorithm color.

Figure 6 indicates that MASSA Algorithm generated distributions with medians higher or equal to random sampling, demonstrating a clear advantage in employing MASSA over random sampling. When compared to random sampling, the MASSA sample's lowest and maximum values were also always greater or equivalent. The

MASSA samplings had less variation than the randomly selected samplings, and the MASSA distribution of concordance values was always shifted to higher values than the random distribution. Median of concordances was also shown in **Table 3** and together with **Figure 6** reveals that, when compared to other sampling methods, the proposed algorithm consistently produced equivalent or higher results in terms of median of percentage of concordance for HQSAR descriptors. However, there were two exceptions to this pattern: (1) In the DHFR dataset, Sphere Exclusion outperformed the proposed algorithm, but MASSA outperformed the original article distribution and was comparable to the Kennard-Stone method; (2) In the JEKO-1 dataset, the original distribution outperformed the proposed algorithm, followed by SPXY, while MASSA results were comparable to other sampling techniques. Taking these findings into account, MASSA clearly demonstrated a more balanced overall performance, outperforming the other algorithms in most cases.

Table 3. Concordance (C) percentage of HQSAR descriptor clusters in training and test sets considering the samples selected from the referenced studies.

Dataset	Median C_{MASSA} (%)	$C_{Kennard-Stone}$ (%)	C_{SPXY} (%)	Median $C_{Sphere Exclusion}$ (%)	Median C_{random} (%)	$C_{original}$ (%)
ACE	84.62	61.54	61.54	76.92	69.23	84.62
CB_1	55.56	44.44	38.89	44.44	38.89	55.56
CMC	100.00	100.00	100.00	100.00	100.00	100.00
DHFR	75.00	75.00	87.50	100.00	87.50	62.50
GPB	64.29	42.86	50.00	50.00	57.14	64.29
JEKO-1	28.57	28.57	35.71	28.57	28.57	50.00
THR	44.44	27.78	33.33	38.89	38.89	44.44

The domain of applicability for RF-QSAR was calculated in a comparable way to that used for HQSAR concordance. However, in this case, the same fingerprint was employed both as a descriptor for model construction and for the domain of applicability assessment. It is worth noting that attaining HQSAR concordance is more challenging than achieving RF-QSAR concordance. This is because the descriptors utilized in the HQSAR method are not accessible prior to model construction, which makes it more challenging to create accurate representations of the chemical space used for model training. This characteristic was also observed in **Table 4**, where the medians of concordance for all methods, except for some original distributions, are 100%. **Figure 7** shows this same pattern; however, we can see the distributions across replicates in greater detail. It is noteworthy that the Sphere Exclusion method underperformed compared to other methods in certain datasets, despite their equivalent medians. Additionally, the observed trend of MASSA performing on par with random in the JEKO-1 dataset can be attributed to its low molecule count (only 35 molecules in total). This makes it challenging to apply rational splitting using the default configuration of MASSA and, overall, difficult to construct QSAR models with robust generalization capabilities. Other linkage methods and a reduction in the percentage of variability explained by PCA can be used to overcome this challenge using MASSA; however, preparing QSAR models with a small number of molecules requires algorithms tailored to this task.

Table 4. Concordance (C) percentage of RF-QSAR descriptors in training and test sets considering the samples selected from the referenced studies.

Dataset	**Median C_{MASSA} (%)**	**$C_{Kennard-Stone}$ (%)**	**C_{SPXY} (%)**	**Median $C_{Sphere Exclusion}$ (%)**	**Median C_{random} (%)**	**$C_{original}$ (%)**
ACE	100.00	100.00	100.00	100.00	100.00	97.37
CB_1	100.00	100.00	100.00	100.00	100.00	100.00
CMC	100.00	100.00	100.00	100.00	100.00	100.00
DHFR	100.00	100.00	100.00	100.00	100.00	98.39
GPB	100.00	100.00	100.00	100.00	100.00	100.00
JEKO-1	100.00	100.00	100.00	100.00	100.00	100.00
THR	100.00	100.00	100.00	100.00	100.00	100.00

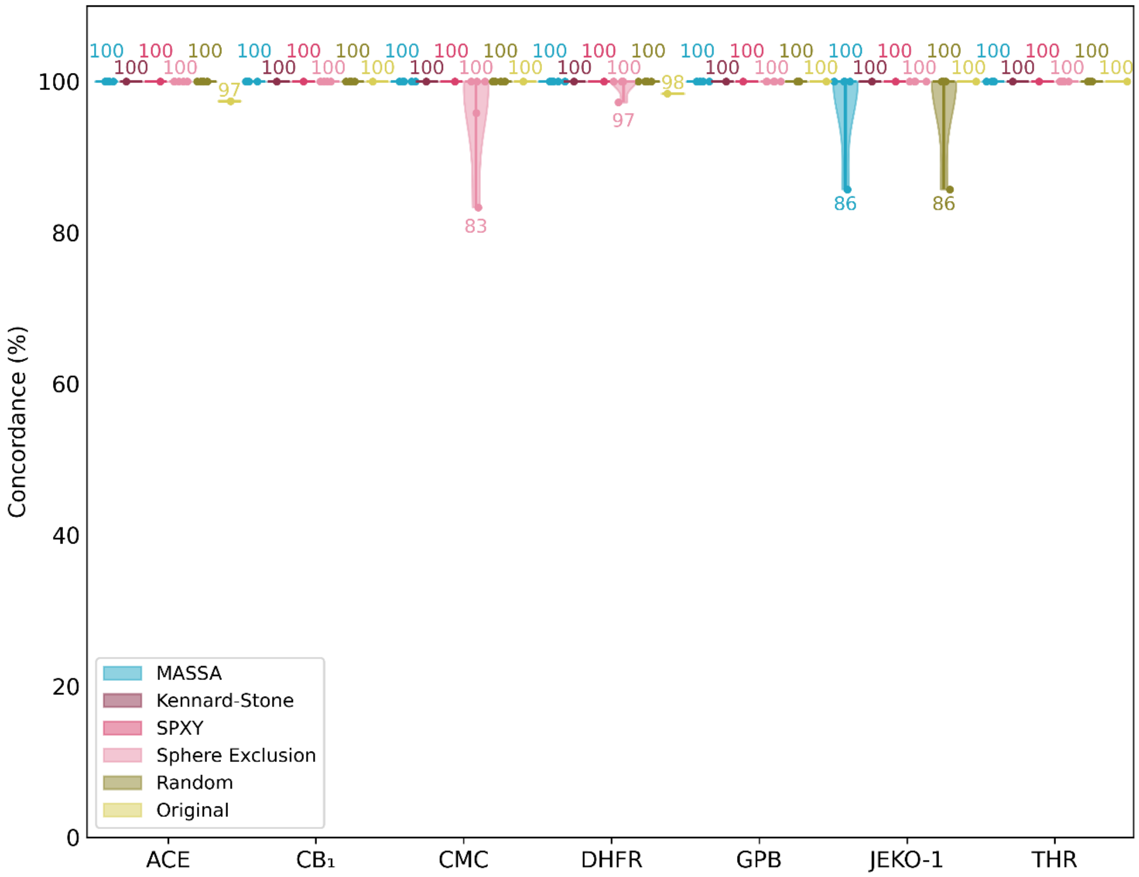

Figure 7. Distribution of the percentage of concordance for the RF-QSAR descriptors across the training and test sets in different datasets. The text values represent the lowest and highest concordance percentages, while the median is represented by a line of the respective sampling algorithm color.

Validation of QSAR models

Moving on to the analysis of validation metrics, **Figure 8** and **Figure 9** shows the variation of the metrics obtained for the replicates of the HQSAR and RF-QSAR models, respectively. The results indicate what has already been discovered in the literature: randomly sampled datasets can produce models with good predictive ability; however, discrepancies in validation metrics were observed, implying that random sampling can affect predictive ability, leading to incorrect conclusions, because the values may be too optimistic or too pessimistic [3, 6, 10]. This is most noticeable in the JEKO-1 and CMC datasets for both QSAR methods and in TRα in RF-QSAR, but in all datasets, the metric values had more variations across replicas for models constructed with the randomly sampled sets. The less significant difference between maximum and minimum values in the validation metrics for the models with the MASSA samples suggests that, when compared to random sampling, the proposed algorithm reduces the risk of building overly optimistic or overly pessimistic models.

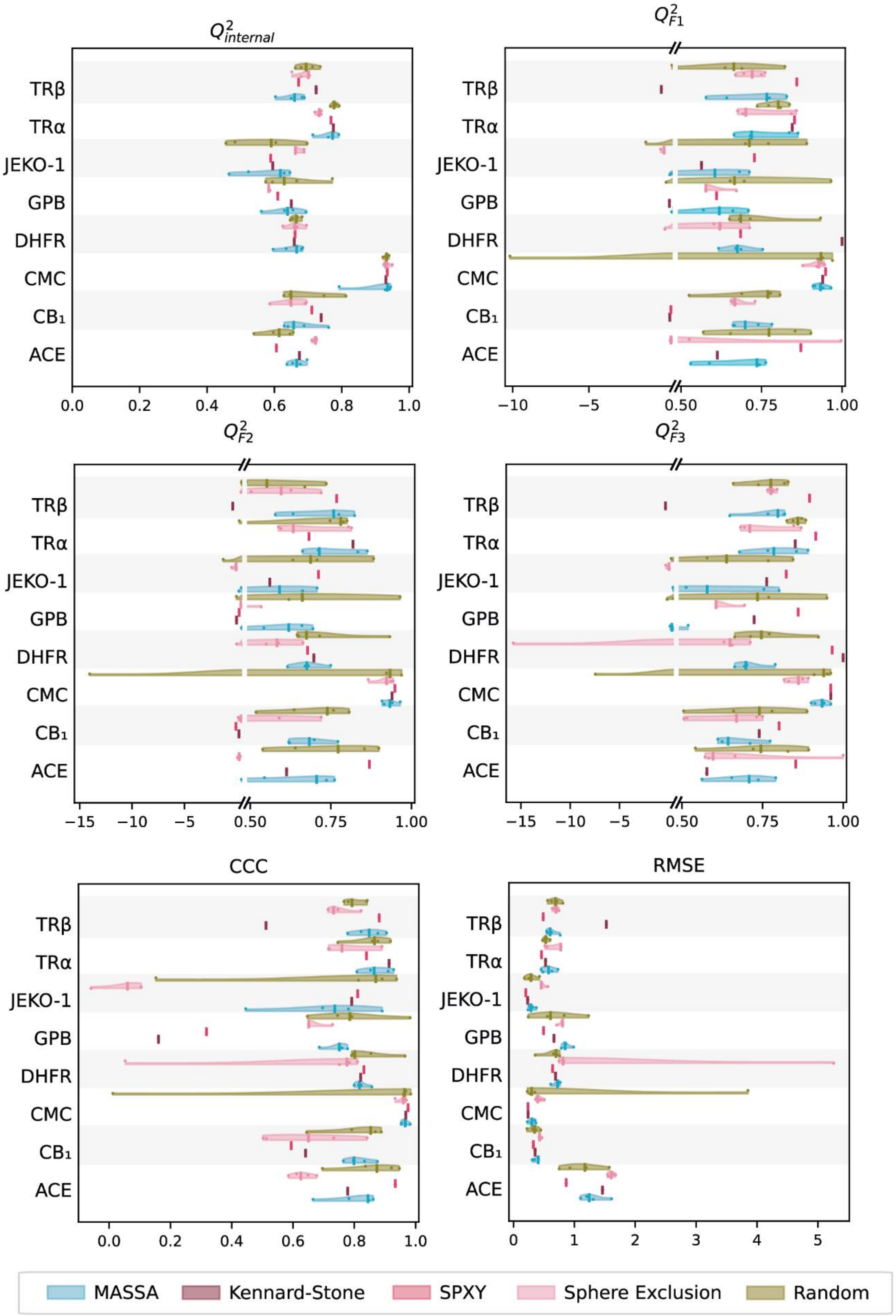

Figure 8. Validation metrics of HQSAR models for the different datasets and sample distributions.

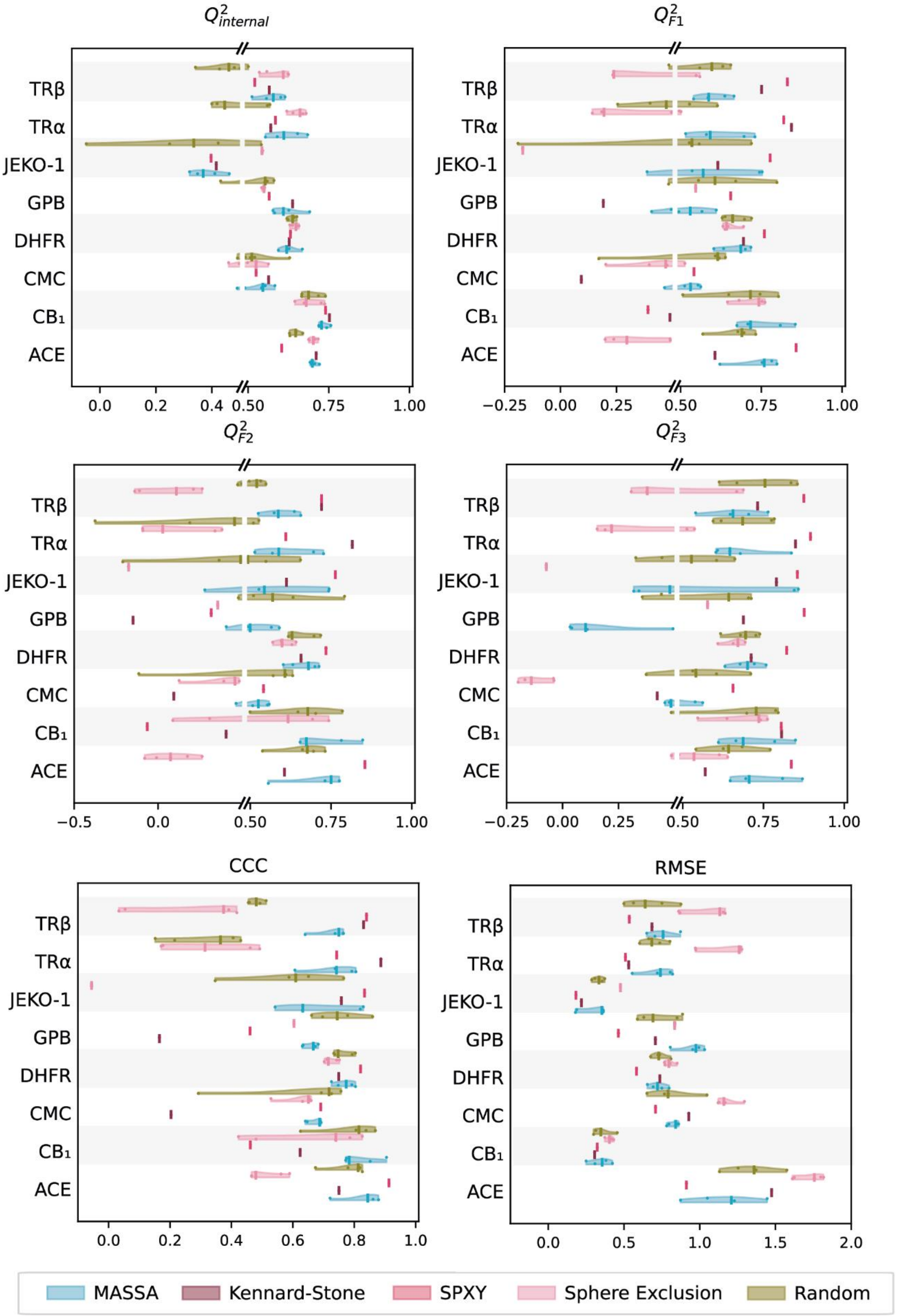

Figure 9. Validation metrics of RF-QSAR models for the different datasets and sample distributions.

MASSA algorithm was also more capable of achieving acceptable values of validation metrics than other rational sampling algorithms, showing that the algorithm is more able to generate predictive QSAR models. This is evident in the violin plots (**Figure 8** and **Figure 9**), where MASSA frequently yielded higher values for $Q^2_{internal}$, Q^2_{F1}, Q^2_{F2}, Q^2_{F3}, and notably CCC when compared to the other methods. Its distributions more often fell within the portion of plots with values surpassing 0.5 across datasets. Additionally, MASSA yielded lower RMSE values, further affirming its superior accuracy performance compared to other sampling strategies. It is interesting to note that Sphere Exclusion performed poorly in the CCC metric for both HQSAR and RF-QSAR, achieving the lowest values observed and unsuitable for QSAR modeling. This sampling method also yielded low values for other validation metrics in RF-QSAR and was the one that generally performed the worst. Furthermore, with an expected reference value being 0.6 or higher [54–57], MASSA was the only sampling algorithm capable of producing usable RF-QSAR models considering CCC metric. These findings suggest that, even when the applicability domain and descriptor concordance are comparable between sampling methods, MASSA was able to achieve better validation metrics due to its ability to cover the chemical space with more information. The extraction of additional information from MASSA relies solely on the molecule's structure and biological activities, both of which are available in every QSAR study, providing a significant advantage.

Despite not covering the entire biological and chemical diversity of the JEKO-1 dataset, the MASSA distribution showed less variation in internal and external validation metrics and a higher level of concordance of the HQSAR descriptors when compared to the random distribution. In this specific dataset, all rational sampling algorithms struggle to find predictive models. MASSA sampling produced predictive HQSAR models, but, like all other sampling methods, it was unable to produce acceptable RF-QSAR models yielding values that were outside of what was expected for the validation metrics. Small molecule datasets often present challenges for QSAR studies. They are frequently unsuitable for QSAR model generation and generally have limited generalization ability. However, they can be a crucial approach when faced with a scarcity of data to discover novel bioactive molecules. In these cases, it is important to approach models derived from these datasets with caution in terms of applicability and generalization. Nevertheless, there are alternative strategies to effectively address this challenge and obtain predictive QSAR models. Examples include "Small Data Set QSAR Modeling," which finds predictive models by using exhaustive cross-validation across different sampling replicates, oversampling strategies, and the use of other machine learning methods based on transfer and few-shot learning [59–61].

Finally, the disparity in the values of the external validation metrics for the random sampled CMC dataset is noteworthy, particularly in HQSAR studies. While all the random and MASSA replicates exhibited perfect concordance for HQSAR and RF-QSAR descriptors, only MASSA replicates covered the entire range of biological activity. **Figure 10** shows the results of the frequency of molecules in clusters within the biological domain for each randomly sampled dataset. The significant drop in the external validation metrics was observed in the fourth random sampling replicate (frequency of the molecules in this replicate are shown in **Figure 10D**). Among all the samples, this replicate was the only one where a cluster with more than 5% representation in the entire dataset was not represented in the test set. This highlights the importance of sampling training and test sets in accordance with the distribution of the dependent variable (biological activity) and shows the direct impact of absence of representation to the external validation. Furthermore, considering that this property range is also overestimated in the training set (all samples within this range are only present in the training data), this occurrence may be attributed to potential overfitting. This observation further justifies the fact that internal validation was not significantly affected, in contrast to the notable impact observed in the external validation. Additionally, overfitting can significantly compromise the applicability domain for future applications, making models developed with this randomly sampled dataset unsuitable for QSAR studies or virtual screenings. In conclusion, this example draws attention to the importance of sampling training and test sets appropriately and rationally. It also highlights the importance of extensively sampling across the physicochemical, structural, and biological spaces to ensure the development of models suitable for QSAR studies, with high predictive ability and inside of applicability domain for the entire chemical and biological spaces. Therefore, it is recommended to employ rational sampling strategies, such as the one presented in this algorithm, to prevent occurrences like this last example.

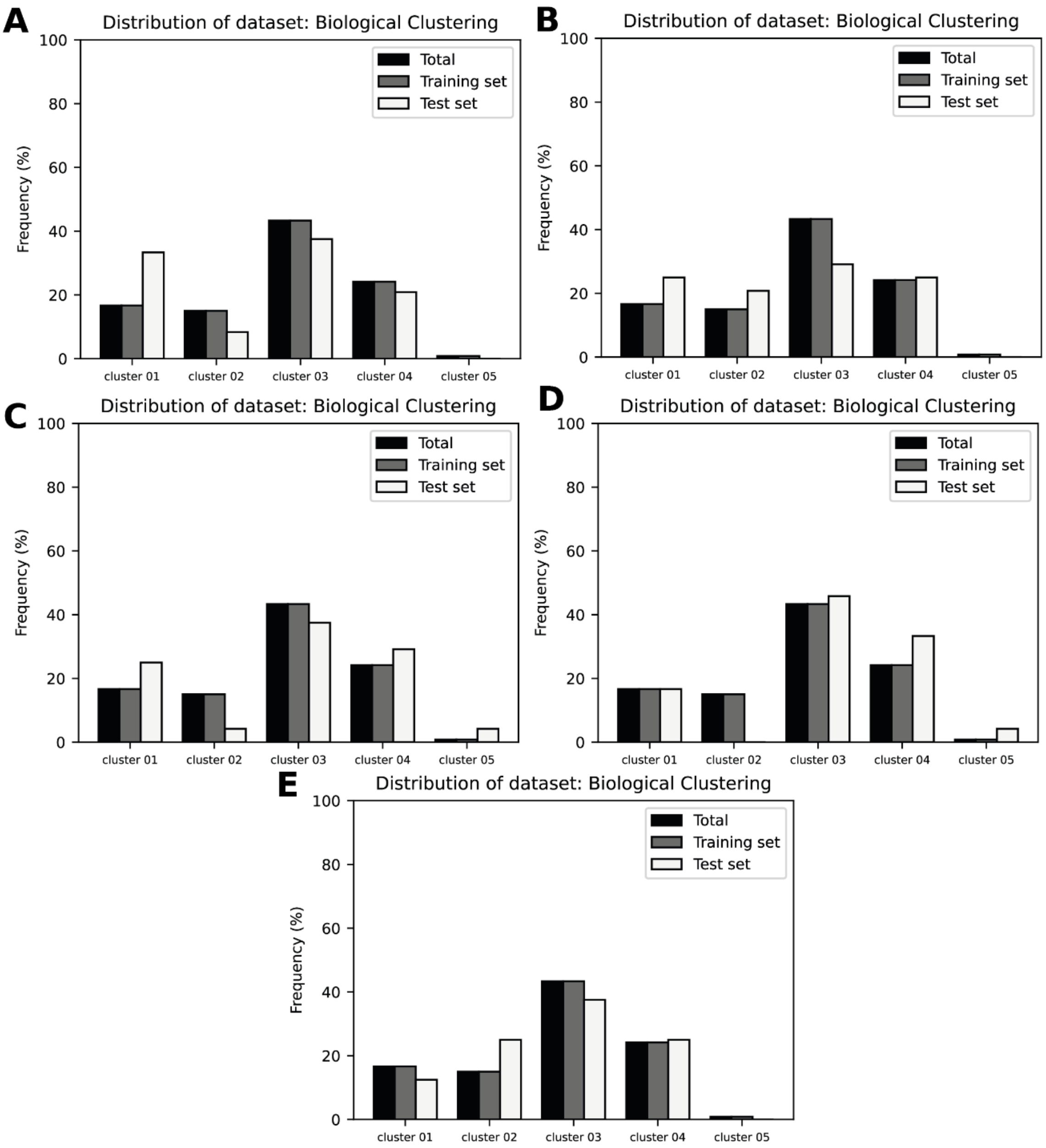

Figure 10. A bar plot showing the frequency of molecules (at the CMC dataset) in each cluster for the biological domain in the first (A), second (B), third (C), fourth (D), and fifth (E) random sampling replicate.

CONCLUSION

In this paper, we proposed MASSA Algorithm, a new tool for the rational selection of training and test data. We based the approach on using the biological activity of compounds combined with other attributes such as Atom Pairs fingerprint, and physicochemical properties, for use by clustering algorithms and principal component analysis during the training and test data split. When compared to random sampling, the proposed tool demonstrated models with reduced variability and better values across multiple replicates for these validation metrics, which also represents a tendency to avoid pessimistic models with inappropriate values for these metrics. Additionally, MASSA frequently yielded higher values for $Q^2_{internal}$, Q^2_{F1}, Q^2_{F2}, Q^2_{F3}, and notably CCC when compared to the other sampling methods, like Kennard-Stone, SPXY and Sphere Exclusion. These results were obtained even when the descriptors used in the QSAR/QSPR were different from those used in the separation of training and test sets, indicating that this tool can be used to build models for more than one QSAR/QSPR technique or with inaccessible X-variables. Finally, as demonstrated in the results section, our methodology not only achieved great training and test data split results, but it also represents an efficient method to sample datasets for generating predictive QSAR/QSPR models with useful visual representations of the distribution.

REFERENCES

1. Yang X, Wang Y, Byrne R, et al (2019) Concepts of Artificial Intelligence for Computer-Assisted Drug Discovery. Chem Rev 119:10520–10594. https://doi.org/10.1021/acs.chemrev.8b00728

2. Masand VH, Mahajan DT, Nazeruddin GM, et al (2015) Effect of information leakage and method of splitting (rational and random) on external predictive ability and behavior of different statistical parameters of QSAR model. Med Chem Res 24:1241–1264. https://doi.org/10.1007/s00044-014-1193-8

3. Andrada MF, Vega-Hissi EG, Estrada MR, Garro Martinez JC (2017) Impact assessment of the rational selection of training and test sets on the predictive ability of QSAR models. SAR and QSAR in Environmental Research 28:1011–1023. https://doi.org/10.1080/1062936X.2017.1397056

4. Clark DE (2006) What has computer-aided molecular design ever done for drug discovery? Expert Opinion on Drug Discovery 1:103–110. https://doi.org/10.1517/17460441.1.2.103

5. International Council for Harmonisation of Technical Requirements for Pharmaceuticals for Human Use (2017) Assessment and Control of DNA Reactive (Mutagenic) Impurities in Pharmaceuticals to Limit Potential Carcinogenic Risk

6. Martin TM, Harten P, Young DM, et al (2012) Does Rational Selection of Training and Test Sets Improve the Outcome of QSAR Modeling? J Chem Inf Model 52:2570–2578. https://doi.org/10.1021/ci300338w

7. Cherkasov A, Muratov EN, Fourches D, et al (2014) QSAR Modeling: Where Have You Been? Where Are You Going To? J Med Chem 57:4977–5010. https://doi.org/10.1021/jm4004285

8. Muratov EN, Bajorath J, Sheridan RP, et al (2020) QSAR without borders. Chem Soc Rev 49:3525–3564. https://doi.org/10.1039/D0CS00098A

9. Puzyn T, Mostrag-Szlichtyng A, Gajewicz A, et al (2011) Investigating the influence of data splitting on the predictive ability of QSAR/QSPR models. Struct Chem 22:795–804. https://doi.org/10.1007/s11224-011-9757-4

10. Esbensen KH, Geladi P (2010) Principles of Proper Validation: use and abuse of re-sampling for validation. Journal of Chemometrics 24:168–187. https://doi.org/10.1002/cem.1310

11. Hawkins DM, Basak SC, Mills D (2003) Assessing Model Fit by Cross-Validation. J Chem Inf Comput Sci 43:579–586. https://doi.org/10.1021/ci025626i

12. Golbraikh A, Tropsha A (2000) Predictive QSAR modeling based on diversity sampling of experimental datasets for the training and test set selection. Mol Divers 5:231–243. https://doi.org/10.1023/A:1021372108686

13. Golbraikh A, Shen M, Xiao Z, et al (2003) Rational selection of training and test sets for the development of validated QSAR models. J Comput Aided Mol Des 17:241–253. https://doi.org/10.1023/A:1025386326946

14. Wu W, Walczak B, Massart DL, et al (1996) Artificial neural networks in classification of NIR spectral data: Design of the training set. Chemometrics and Intelligent Laboratory Systems 33:35–46. https://doi.org/10.1016/0169-7439(95)00077-1

15. Kronenberger T, Windshügel B, Wrenger C, et al (2018) On the relationship of anthranilic derivatives structure and the FXR (Farnesoid X receptor) agonist activity. Journal of Biomolecular Structure and Dynamics 36:4378–4391. https://doi.org/10.1080/07391102.2017.1417161

16. Veríssimo GC, Menezes Dutra EF, Teotonio Dias AL, et al (2019) HQSAR and random forest-based QSAR models for anti-T. vaginalis activities of nitroimidazoles derivatives. Journal of Molecular Graphics and Modelling 90:180–191. https://doi.org/10.1016/j.jmgm.2019.04.007

17. Gomes RA, Genesi GL, Maltarollo VG, Trossini GHG (2017) Quantitative structure–activity relationships (HQSAR, CoMFA, and CoMSIA) studies for COX-2 selective inhibitors. Journal of Biomolecular Structure and Dynamics 35:1436–1445. https://doi.org/10.1080/07391102.2016.1185379

18. Fernandes P de O, Martins JPA, Melo EB de, et al (2021) Quantitative structure-activity relationship and machine learning studies of 2-thiazolylhydrazone derivatives with anti-Cryptococcus neoformans activity. Journal of Biomolecular Structure and Dynamics 0:1–12. https://doi.org/10.1080/07391102.2021.1935321

19. Kronenberger T, Asse LR, Wrenger C, et al (2017) Studies of Staphylococcus aureus FabI inhibitors: fragment-based approach based on holographic structure–activity relationship analyses. Future Medicinal Chemistry 9:135–151. https://doi.org/10.4155/fmc-2016-0179

20. Ferreira GM, Magalhães JG de, Maltarollo VG, et al (2020) QSAR studies on the human sirtuin 2 inhibition by non-covalent 7,5,2-anilinobenzamide derivatives.

Journal of Biomolecular Structure and Dynamics 38:354–363. https://doi.org/10.1080/07391102.2019.1574603

21. Maltarollo VG (2019) Classification of Staphylococcus Aureus FabI Inhibitors by Machine Learning Techniques. IJQSPR 4:1–14. https://doi.org/10.4018/IJQSPR.2019100101

22. Primi MC, Maltarollo VG, Magalhães JG, et al (2016) Convergent QSAR studies on a series of NK3 receptor antagonists for schizophrenia treatment. Journal of Enzyme Inhibition and Medicinal Chemistry 31:283–294. https://doi.org/10.3109/14756366.2015.1021250

23. Popova M, Isayev O, Tropsha A (2018) Deep reinforcement learning for de novo drug design. Science Advances 4:eaap7885. https://doi.org/10.1126/sciadv.aap7885

24. Schneider G (2019) Mind and machine in drug design. Nat Mach Intell 1:128–130. https://doi.org/10.1038/s42256-019-0030-7

25. Dara S, Dhamercherla S, Jadav SS, et al (2022) Machine Learning in Drug Discovery: A Review. Artif Intell Rev 55:1947–1999. https://doi.org/10.1007/s10462-021-10058-4

26. Ambure P, Halder AK, González Díaz H, Cordeiro MNDS (2019) QSAR-Co: An Open Source Software for Developing Robust Multitasking or Multitarget Classification-Based QSAR Models. J Chem Inf Model 59:2538–2544. https://doi.org/10.1021/acs.jcim.9b00295

27. Halder AK, Dias Soeiro Cordeiro MN (2021) QSAR-Co-X: an open source toolkit for multitarget QSAR modelling. Journal of Cheminformatics 13:29. https://doi.org/10.1186/s13321-021-00508-0

28. Veríssimo GC (2021) MASSA Algorithm: Molecular data set sampling for training-test separation

29. Landrum G (2021) RDkit: 2021_03_3 (Q1 2021) Release

30. Vos NJ de (2015) KModes categorical clustering library

31. Python Software Foundation argparse — Parser for command-line options, arguments and sub-commands — Python 3.9.7 documentation. https://docs.python.org/3/library/argparse.html. Accessed 5 Oct 2021

32. scikit-learn: machine learning in Python — scikit-learn 1.0 documentation. https://scikit-learn.org/stable/index.html. Accessed 5 Oct 2021

33. sklearn.decomposition.PCA. In: scikit-learn. https://scikit-learn/stable/modules/generated/sklearn.decomposition.PCA.html. Accessed 5 Oct 2021

34. scipy.cluster.hierarchy.linkage — SciPy v1.7.1 Manual. https://docs.scipy.org/doc/scipy/reference/generated/scipy.cluster.hierarchy.linkage.html. Accessed 8 Oct 2021

35. scipy.cluster.hierarchy.maxdists — SciPy v1.8.0 Manual. https://docs.scipy.org/doc/scipy/reference/generated/scipy.cluster.hierarchy.maxdists.html. Accessed 22 Mar 2022

36. scipy.cluster.hierarchy.fcluster — SciPy v1.7.1 Manual. https://docs.scipy.org/doc/scipy/reference/generated/scipy.cluster.hierarchy.fcluster.html. Accessed 8 Oct 2021

37. scipy.cluster.hierarchy.dendrogram — SciPy v1.7.1 Manual. https://docs.scipy.org/doc/scipy/reference/generated/scipy.cluster.hierarchy.dendrogram.html. Accessed 8 Oct 2021

38. sklearn.model_selection.train_test_split. In: scikit-learn. https://scikit-learn/stable/modules/generated/sklearn.model_selection.train_test_split.html. Accessed 9 Oct 2021

39. Sutherland JJ, O'Brien LA, Weaver DF (2004) A Comparison of Methods for Modeling Quantitative Structure−Activity Relationships. J Med Chem 47:5541–5554. https://doi.org/10.1021/jm0497141

40. Liu C-J, Yu S-L, Liu Y-P, et al (2016) Synthesis, cytotoxic activity evaluation and HQSAR study of novel isosteviol derivatives as potential anticancer agents. European Journal of Medicinal Chemistry 115:26–40. https://doi.org/10.1016/j.ejmech.2016.03.009

41. Valadares NF, Castilho MS, Polikarpov I, Garratt RC (2007) 2D QSAR studies on thyroid hormone receptor ligands. Bioorganic & Medicinal Chemistry 15:4609–4617. https://doi.org/10.1016/j.bmc.2007.04.015

42. Ye M, Dawson MI (2009) Studies of cannabinoid-1 receptor antagonists for the treatment of obesity: Hologram QSAR model for biarylpyrazolyl oxadiazole ligands. Bioorganic & Medicinal Chemistry Letters 19:3310–3315. https://doi.org/10.1016/j.bmcl.2009.04.072

43. Jiao L, Wang Y, Qu L, et al (2020) Hologram QSAR study on the critical micelle concentration of Gemini surfactants. Colloids and Surfaces A: Physicochemical and Engineering Aspects 586:124226. https://doi.org/10.1016/j.colsurfa.2019.124226

44. Dassault Systèmes Biovia Corp (2020) BIOVIA Discovery Studio Visualizer 2021

45. Hawkins PCD, Skillman AG, Warren GL, et al (2010) Conformer Generation with OMEGA: Algorithm and Validation Using High Quality Structures from the Protein Databank and Cambridge Structural Database. Journal of Chemical Information and Modeling 50:572–584. https://doi.org/10.1021/ci100031x

46. Hawkins, P.C.D. OMEGA. OpenEye Scientific Software, Santa Fe, NM

47. QUACPAC. OpenEye Scientific Software, Santa Fe, NM

48. Burns J, Spiekermann K, Bhattacharjee H, et al (2023) Machine Learning Validation via Rational Dataset Sampling with astartes

49. TRIPOS Associates Inc (2012) Sybyl-X Molecular Modeling Software Packages

50. Berthold MR, Cebron N, Dill F, et al (2009) KNIME - the Konstanz information miner: version 2.0 and beyond. ACM SIGKDD Explorations Newsletter 11:26. https://doi.org/10.1145/1656274.1656280

51. Fernandes PO, Martins DM, de Souza Bozzi A, et al (2021) Molecular insights on ABL kinase activation using tree-based machine learning models and molecular docking. Mol Divers 25:1301–1314. https://doi.org/10.1007/s11030-021-10261-z

52. Pedregosa F, Varoquaux G, Gramfort A, et al (2011) Scikit-learn: Machine learning in Python. The Journal of Machine Learning Research 12:2825–2830

53. Virtanen P, Gommers R, Oliphant TE, et al (2020) SciPy 1.0: fundamental algorithms for scientific computing in Python. Nat Methods 17:261–272. https://doi.org/10.1038/s41592-019-0686-2

54. Chirico N, Gramatica P (2011) Real External Predictivity of QSAR Models: How To Evaluate It? Comparison of Different Validation Criteria and Proposal of Using the Concordance Correlation Coefficient. Journal of Chemical Information and Modeling 51:2320–2335. https://doi.org/10.1021/ci200211n

55. Golbraikh A, Tropsha A (2002) Beware of q2! Journal of Molecular Graphics and Modelling 20:269–276. https://doi.org/10.1016/S1093-3263(01)00123-1

56. Roy K, Kar S, Das RN (2015) A Primer on QSAR/QSPR Modeling. Springer International Publishing, Cham

57. Shi LM, Fang H, Tong W, et al (2001) QSAR Models Using a Large Diverse Set of Estrogens. Journal of Chemical Information and Computer Sciences 41:186–195. https://doi.org/10.1021/ci000066d

58. Gramatica P, Sangion A (2016) A Historical Excursus on the Statistical Validation Parameters for QSAR Models: A Clarification Concerning Metrics and Terminology. J Chem Inf Model 56:1127–1131. https://doi.org/10.1021/acs.jcim.6b00088

59. Bae S-Y, Lee J, Jeong J, et al (2021) Effective data-balancing methods for class-imbalanced genotoxicity datasets using machine learning algorithms and molecular fingerprints. Computational Toxicology 20:100178. https://doi.org/10.1016/j.comtox.2021.100178

60. Veríssimo GC, Serafim MSM, Kronenberger T, et al (2022) Designing drugs when there is low data availability: one-shot learning and other approaches to face the issues of a long-term concern. Expert Opinion on Drug Discovery 17:929–947. https://doi.org/10.1080/17460441.2022.2114451

61. Ambure P, Gajewicz-Skretna A, Cordeiro MNDS, Roy K (2019) New Workflow for QSAR Model Development from Small Data Sets: Small Dataset Curator and Small Dataset Modeler. Integration of Data Curation, Exhaustive Double Cross-Validation, and a Set of Optimal Model Selection Techniques. J Chem Inf Model 59:4070–4076. https://doi.org/10.1021/acs.jcim.9b00476