QuBiLS-MIDAS: A Parallel Free-Software for Molecular Descriptors Computation Based on Multilinear Algebraic Maps

César R. García-Jacas,[a,b] Yovani Marrero-Ponce,*[b,c,d] Liesner Acevedo-Martínez,[a] Stephen J. Barigye,[b,e] José R. Valdés-Martiní,[b,f] and Ernesto Contreras-Torres[a]

The present report introduces the QuBiLS-MIDAS software belonging to the ToMoCoMD-CARDD suite for the calculation of three-dimensional molecular descriptors (MDs) based on the two-linear (bilinear), three-linear, and four-linear (multilinear or *N*-linear) algebraic forms. Thus, it is unique software that computes these tensor-based indices. These descriptors, establish relations for two, three, and four atoms by using several (dis-)similarity metrics or multimetrics, matrix transformations, cutoffs, local calculations and aggregation operators. The theoretical background of these *N*-linear indices is also presented. The QuBiLS-MIDAS software was developed in the Java programming language and employs the Chemical Development Kit library for the manipulation of the chemical structures and the calculation of the atomic properties. This software is composed by a desktop user-friendly interface and an Abstract Programming Interface library. The former was created to simplify the configuration of the different options of the MDs, whereas the library was designed to allow its easy integration to other software for chemoinformatics applications. This program provides functionalities for data cleaning tasks and for batch processing of the molecular indices. In addition, it offers parallel calculation of the MDs through the use of all available processors in current computers. The studies of complexity of the main algorithms demonstrate that these were efficiently implemented with respect to their trivial implementation. Lastly, the performance tests reveal that this software has a suitable behavior when the amount of processors is increased. Therefore, the QuBiLS-MIDAS software constitutes a useful application for the computation of the molecular indices based on *N*-linear algebraic maps and it can be used freely to perform chemoinformatics studies. © 2014 Wiley Periodicals, Inc.

DOI: 10.1002/jcc.23640

Introduction

Molecular descriptors (MDs) are numeric values obtained as result of mathematical transformations of molecular structure representations.[1] These are computed from different theories and are widely used in different studies and scientific fields, such as: quantitative structure-activity relationships/quantitative structure-property relationships (QSAR/QSPR) studies, similarity/dissimilarity studies, absorption, distribution, metabolism, elimination - toxicity (ADME-Tox) applications, and so on. It is obvious that a single descriptor may not suffice to solely characterize all molecular structural features. In this way, it is always important to consider a wide space of MDs to adequately codify chemical information contained in molecular representations.

Currently, there are several commercial- and free-software and libraries that have been developed to calculate part of the existing MDs according to criteria followed by their authors or include the calculation of MDs as one of its features.[2–9] The main purpose of all these informatics applications is to provide a tool that facilitates the configuration of MDs for their calculation. These computational programs, in general sense, are capable of calculating various kinds of molecular indices, can run in different architectures and operative systems, and have a graphic user interface and/or an interface based on command line to interact with the final user. These aspects previously mentioned constitute desirable featurees that MD calculating software should possess.

However, the most of these software present some limitations or aspects to improve, which must be taken into account in the development of future applications, such as:

[a] César R. García-Jacas, L. Acevedo-Martínez, E. Contreras-Torres
Grupo de Investigación de Bioinformática, Centro de Estudio de Matemática Computacional, Universidad de las Ciencias Informáticas, La Habana, Cuba

[b] César R. García-Jacas, Y. Marrero-Ponce, S. J. Barigye, José R. Valdés-Martiní
Unit of Computer-Aided Molecular "Biosilico" Discovery and Bioinformatic Research (CAMD-BIR Unit), Faculty of Chemistry-Pharmacy, Universidad Central "Martha Abreu" de Las Villas, Santa Clara 54830, Villa Clara, Cuba
E-mail: ymarrero77@yahoo.es or ymponce@gmail.com

[c] Y. Marrero-Ponce
Institut Universitari de Ciència Molecular, Universitat de València, Edifici d'Instituts de Paterna, P.O. Box 22085, València E-46071, Spain

[d] Y. Marrero-Ponce
Facultad de Química Farmacéutica, Universidad de Cartagena, Cartagena de Indias, Bolívar, Colombia

[e] S. J. Barigye
School of Pharmacy, Kampala International University, Western Campus, P.O. Box 71, Bushenyi, Uganda

[f] José R. Valdés-Martiní
Facultad de Matemática, Centro de Estudios de Informática (CEI), Física y Computación, Universidad Central "Marta Abreu" de Las Villas, Santa Clara, Villa Clara 54830, Cuba

should be fully cross-platform; should be implemented to perform multicore processing; should perform batch processing of MDs; should have procedures for performing cleaning and curation tasks of molecular datasets; should compute local indices and so be used to analyze determined chemical-fragments or atom-types; should allow considering the lone-pair electrons of the atoms, as well as, to add (or remove) hydrogen atoms for the computation of molecular indices. In addition, the use of cutoffs for the analysis of the most important noncovalent or covalent interactions (short-, middle-, and large-contacts), the use of probabilistic transformations in the matrix representations and the use of mathematical operators (so called aggregation operators) different from the summation to obtain total (or local) MDs from atomic contributions [local vertex invariants (LOVIs)], are features rarely used in current software. Lastly, none of the existing programs include theoretical aspects to compute molecular indices based on relations for more than two atoms, for example, using multimetrics (tensor of degree 3 or 4).

In recent reports,[10,11] Marrero-Ponce et al. inspired by the successful results achieved by the application of the algebraic two-dimensional (2D)-MDs,[12–21] have proposed the QuBiLS-MIDAS [acronym for Quadratic, Bilinear, and *N*-Linear Maps based on *N*-tuple Spatial Metric [(Dis)-Similarity] Matrices and Atomic Weightings] indices to consider geometric [three-dimensional (3D)] features. These novel molecular parameters utilize the bilinear, quadratic and linear algebraic forms to codify the 3D-information of chemical structures using several (dis-)similarity metrics when analysis between atom-pairs is performed. In addition, these molecular parameters use the *N*-linear ($N > 2$) algebraic maps (as generalized mathematical expressions of the bilinear, quadratic, and linear algebraic forms) to take into account information between three and four atoms of the molecules by means of (dis-)similarity multimetrics. This last approach has never been used before in the definition of MDs.

These indices were assessed in three different chemoinformatics studies: (1) variability analysis based on Shannon's entropy,[21,22] (2) linear-independence of the codified information using Principal Component Analysis,[23] and (3) correlation studies with determined biological activity (QSAR). For the studies (1) and (2), the comparisons were performed with respect to the geometric indices calculated by the DRAGON software,[4] whereas in study (3) the results were analyzed with respect to 3D-QSAR methodologies using Cramer's steroids dataset[24] and in other eight databases studied by Sutherland et.al.[25] In all cases, the results achieved have a comparable-to-superior behavior according to the comparison criterion followed (García-Jacas et al., in process).[10,11]

In this way and taking into consideration the good results attained, it is necessary the design of a computer program which contribute to the calculation of these indices and therefore, this manuscript is dedicated to the presentation of the free, cross-platform and parallel QuBiLS-MIDAS software belonging to the TOMOCOMD-CARDD [acronym for Topological Molecular Computational Design – Computed Aided Rational Drug Design] framework.

QUBILS-MIDAS 3D-Descriptors

Mathematical definition for the *N*-linear form-based algebraic indices

The QuBiLS-MIDAS molecular 3D (geometric) indices[10,11] (see Supporting Information 1) are computed from the atomic contribution of each atom in a molecule. In this way, if a molecule consists of *n* atoms, then the *k*th two-linear (bilinear, quadratic, and linear), three-linear (threelinear, threelinear-quadratic-bilinear, threelinear-bilinear, threelinear-linear, and threelinear-cubic), and four-linear (fourlinear, fourlinear-quadratic-threelinear, fourlinear-threelinear, fourlinear-cubic-bilinear, fourlinear-bilinear, fourlinear-linear, and fourlinear-quadruple) indices for atom "*a*" are calculated as *N*-linear algebraic maps[26,27] (forms) in $\mathbb{R}^n$, in a canonical basis set, and are expressed by the following equations, respectively:

$$ {}_{b}L_a = b^{a,k}(\bar{x},\bar{y}) = \sum_{i=1}^{n}\sum_{j=1}^{n} g_{ij}^{a,k} x^i y^j = [X]^T \mathbb{G}^{a,k}[Y] \quad \forall\ a=1,2,\ldots,n \tag{1} $$

$$ {}_{tr}L_a = tr^{a,k}(\bar{x},\bar{y},\bar{z}) = \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{l=1}^{n} gt_{ijl}^{a,k} x^i y^j z^l = \mathbb{GT}^{a,k}\cdot\bar{x}\cdot\bar{y}\cdot\bar{z} \tag{2} $$

$$ \begin{aligned} {}_{qu}L_a &= qu^{a,k}(\bar{x},\bar{y},\bar{z},\bar{w}) = \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{l=1}^{n}\sum_{h=1}^{n} gq_{ijl}^{a,k} x^i y^j z^l w^h \\ &= \mathbb{GQ}^{a,k}\cdot\bar{x}\cdot\bar{y}\cdot\bar{z}\cdot\bar{w} \end{aligned} \tag{3} $$

where *k* is the power to which the matrix representation is raised (see *N*-tuples spatial-(dis) similarity matrices to represent 3D-information of the chemical structures section), "*a*" indicates the atom ($a=1,2,\ldots,n$), *n* is the number of atoms in a molecule, L_a is the entry corresponding to the contribution of the atom "*a*" in the vector of atom-level indices $\bar{L}$ (designated here by the well-known acronym: LOVI)[28,29] and $x^1,\ldots,x^n$, $y^1,\ldots,y^n$, $z^1,\ldots,z^n$ and $w^1,\ldots,w^n$ are the coordinates or components of the molecular vectors $\bar{x}$, $\bar{y}$, $\bar{z}$, and $\bar{w}$ in a system of canonical (natural) basis vectors of $\mathbb{R}^n$.

The atomic properties to represent the chemical structures have been oftenly used in several reports,[15,16,20,30] and as can be observed in eqs. (1)–(3) several combinations according to the different values of the vectors $\bar{x}$, $\bar{y}$, $\bar{z}$, and $\bar{w}$ are obtained for the two-linear, three-linear, and four-linear algebraic maps. In this way, the algebraic forms shown in the Table 1 are used, and as component of the molecular vectors the following "standard" atom- and fragment-based properties (weights): (1) atomic mass (M), (2) the van der Waals volume (V), (3) the atomic polarizability (P), (4) atomic electronegativity in Pauling scale (E), (5) atomic Ghose-Crippen LogP (A),[31–33] (6) atomic charge (C) (Gasteiger–Marsili),[34] (7) atomic polar surface area,[35] (8) atomic refractivity (R),[31–33] (9) atomic hardness[36] (H), and (10) atomic softness[37] (S).

The coefficients $g_{ij}^{a,k}$[$gt_{ijl}^{a,k}$, $gq_{ijlh}^{a,k}$] are the elements of the *k*th two-tuples [three-tuples, four-tuples] atom-level spatial-(dis)-similarity matrices, $\mathbb{G}^{a,k}$ [$\mathbb{GT}^{a,k}$, $\mathbb{GQ}^{a,k}$] for atom "*a*,"

Table 1. N-linear algebraic forms implemented for the different values of the molecular property vector.

1. Two-linear $\left[\boldsymbol{b}^{\boldsymbol{k}}(\bar{\boldsymbol{x}},\bar{\boldsymbol{y}})\right]$
   - Linear (X, $Y = 1$)
   - Bilinear ($X <> Y$)
   - Quadratic ($X = Y$)
2. Three-linear $\left[\mathbf{tr}^{\boldsymbol{k}}(\bar{\boldsymbol{x}},\bar{\boldsymbol{y}},\bar{\boldsymbol{z}})\right]$
   - Threelinear ($X <> Y <> Z$)
   - Threelinear-quadratic-bilinear ($(X = Y) <> Z$)
   - Threelinear-bilinear ($X <> Y$, $Z = 1$)
   - Threelinear-linear (X, $Y = 1$, $Z = 1$)
   - Threelinear-cubic ($X = Y = Z$)
3. Four-linear $\left[\boldsymbol{qu}^{\boldsymbol{k}}(\bar{\boldsymbol{x}},\bar{\boldsymbol{y}},\bar{\boldsymbol{z}},\bar{\boldsymbol{w}})\right]$
   - Fourlinear ($X <> Y <> Z <> W$)
   - Fourlinear-quadratic-threelinear ($(X = Y) <> Z <> W$)
   - Fourlinear-threelinear ($X = 1$, $Y <> Z <> W$)
   - Fourlinear-cubic-bilinear ($(X = Y = Z) <> W$)
   - Fourlinear-bilinear ($X = Y = 1$, $Z <> W$)
   - Fourlinear-linear ($X = Y = Z = 1$, W)
   - Fourlinear-quadruple ($X = Y = Z = W$)

Symbols Used 1: Using the unitary vector. $<>$: Using different properties. $=$: Using equal properties.

respectively. These atom-level coefficients are obtained from kth two-tuples [three-tuples, four-tuples] total spatial-(dis)similarity matrix, $\mathbb{G}^k$ [$\mathbb{GT}^k$, $\mathbb{GQ}^k$] (see subsection 2 in Supporting Information 1) according to the influence of the atom "*a*" in the relations between two [three, four] atoms of a molecule. So, each atom-level matrix defines an atom-level index for each atom "*a*" [see eqs. (1)–(3)]. The procedures to compute the coefficients $g_{ij}^{a,k}$[$gt_{ijl}^{a,k}$, $gq_{ijlh}^{a,k}$] are specified in the reports[10,11] (see eqs. (4)–(6) in Supporting Information 1).

Therefore, from the previous definitions [see eqs. (1)–(3)], the kth total (whole-molecule) N-linear [where $N = 2$ (two-linear), 3 (three-linear), 4 (four-linear) atoms] indices can be calculated as a linear combination of the atomic contributions (components of vector $\bar{L}$), coinciding these results (molecular indices) with the original approach of the algebraic maps used. However, it has been demonstrated in the reports[38–40] that using other mathematical operators different from sum more suitable results are obtained. These operators, called aggregation operators or invariants (see Supporting Information 2 for mathematical definitions), are also applied to vector $\bar{L}$ to compute the QuBiLS-MIDAS indices. In this way, the attainment of the N-linear indices by summation of the LOVIs is generalized.

N-tuples spatial-(dis) similarity matrices to represent 3D-information of the chemical structures

The codification of 3D information of the chemical structures to compute the proposed indices is performed through the kth two-tuples, three-tuples, and four-tuples spatial-(dis)similarity matrices [$\mathbb{G}^k$, $\mathbb{GT}^k$, and $\mathbb{GQ}^k$] for the duplex, ternary, and quaternary atom relations, respectively [see eqs. (1)–(3)]. The superscript k indicates the power to which $\mathbb{G}$, $\mathbb{GT}$, and $\mathbb{GQ}$ are raised. For $k = 0$, all entries of the matrices $\mathbb{G}^0$, $\mathbb{GT}^0$, and $\mathbb{GQ}^0$ have value 1 and for $k = 1$, the coefficients g_{ij}^1, gt_{ijl}^1, and gq_{ijlh}^1 corresponding to the matrices $\mathbb{G}^1$, $\mathbb{GT}^1$, and $\mathbb{GQ}^1$ provide information on the interactions for two, three, and four atoms, respectively.

The use of the elements of $\mathbb{G}^1$, $\mathbb{GT}^1$, and $\mathbb{GQ}^1$ and their reciprocal matrix for the N-linear indices is inspired in the physicochemical nature of different noncovalent interactions, such as Coulomb potential, gravitational interactions, Van der Waals terms, and so on. In fact, the kth power of $\mathbb{G}^k$, $\mathbb{GT}^k$, and $\mathbb{GQ}^k$ are related with the powers of their elements, where $k = 0$, ±1, ±2, $\pm\pm3$, ..., ±12. These real exponents take into consideration the different noncovalent interactions among atoms in molecule. For instance, for $k = \pm1$ and $k = \pm2$, the $\mathbb{G}^k$, $\mathbb{GT}^k$, and $\mathbb{GQ}^k$ reflect Coulombic-like and/or gravitational-like interactions, respectively. The maximum k value, ±12, is related with the nonbonded (mainly steric) interactions associated with the functional form of the Lennard-Jones 6–12 potential, like in most CoMFA-like studies.

The molecular information is codified by means of (dis)-similarity metrics and multimetrics. In this manner, when interactions based on atom-pairs are considered then several distance measures (metrics) are used (see Table 2),[10] achieving in this way a generalization of the geometrical distance matrix[41] where each entry only corresponds to the Euclidean distance[1],[42–44] between two atoms. In Ref. [10], it was demonstrated that with the use of these metrics different from the Euclidean distance orthogonal information is codified and consequently, better results are achieved in QSAR studies. Conversely, several multimetrics (see Table 3) are used to codify the existing information in relations between three (or four) atoms of a molecule, for example, the triangle area (for three distinct atoms). The objective of these multimetrics is to provide other measures to capture relevant information when n atoms are considered and so, to take into account noncovalent interactions other than the common one between two atoms.[11]

It is important to highlight that the diagonal elements of the matrices $\mathbb{G}^1$, $\mathbb{GT}^1$, and $\mathbb{GQ}^1$ could have values different from zero with the objective of achieving a greater discrimination of the molecular structures. In this sense, the following two options are taken into account: (1) the number of lone-pair electrons for atoms or (2) the spatial distance for each atom i and center of the molecule. This distance is computed with the (dis)-similarity metric(s) [see Table 2] selected when atom-pair relations are analyzed, or when multimetrics depending on the distance between two atoms are employed, for example, Perimeter. For the volume (m23), bond angle (m27–m28), and dihedral angle (m29–m30) multimetrics, if the distance to molecule center is considered then it is calculated with the Euclidean metric.

The calculation of the matrices $\mathbb{G}^k$, $\mathbb{GT}^k$, and $\mathbb{GQ}^k$ for $k \geq 2$ is performed through the Hadamard matrix product and thus these can be considered as generalized matrices.[41] It follows that when the exponent k is negative then the reciprocal of each entry of the N-tuples matrices is computed, except for the diagonal elements when the numbers of lone-pairs is considered.

These previous N-tuples matrices ($\mathbb{G}^k$, $\mathbb{GT}^k$, and $\mathbb{GQ}^k$) can be also used to codify information related with groups or

Table 2. Metrics used to compute the “distance” between two atoms of a molecule.

Metrics	Formula[a]	Range[b]	Average	Range
Minkowski (m1–m7) $p = 0.25, 0.5, 1, 1.5, 2, 2.5, 3$, and ∞ (where, when $p = 1$, it is the Manhattan, city-block, or taxi distance (also known as Hamming distance between binary vectors) and $p = 2$ is Euclidean distance)	$d_{XY} = \left(\sum_{j=1}^{h} \lvert x_j - y_j \rvert^p \right)^{\frac{1}{p}}$	$[0, \infty)$	$\bar{d} = \frac{d_{XY}}{n^{1/p}}$	$[0, \infty)$
Chebyshev/Lagrange (m8) (Minkowski formula when $p = \infty$)	$d_{XY} = \max \left\{ \lvert x_j - y_j \rvert \right\}$			
Canberra (m10)	$d_{XY} = \sum_{j=1}^{h} \frac{\lvert x_j - y_j \rvert}{\lvert x_j \rvert + \lvert y_j \rvert}$	$[0, n]$	$\bar{d} = \frac{d_{XY}}{n}$	$[0, 1]$
Lance–Williams/Bray–Curtis (m11)	$d_{XY} = \frac{\sum_{j=1}^{h} \lvert x_j - y_j \rvert}{\sum_{j=1}^{h} \left(\lvert x_j \rvert + \lvert y_j \rvert \right)}$	$[0, 1]$	$\bar{d} = \frac{d_{XY}}{n}$	$[0, \frac{1}{n}]$
Clark/Coefficient of Divergence (m12)	$d_{XY} = \sqrt{\sum_{j=1}^{h} \left(\frac{x_j - y_j}{\lvert x_j \rvert + \lvert y_j \rvert} \right)^2}$	$[0, n]$	$\bar{d} = \frac{d_{XY}}{\sqrt{n}}$	$[0, \sqrt{n}]$
Soergel (m13)	$d_{XY} = \frac{1}{n} \sum_{j=1}^{h} \frac{\lvert x_j - y_j \rvert}{\max \left\{ x_j, y_j \right\}}$	$[0, 1]$	$\bar{d} = \frac{d_{XY}}{n}$	$[0, \frac{1}{n}]$
Bhattacharyya (m14)	$d_{XY} = \sqrt{\sum_{j=1}^{h} \left(\sqrt{x_j} - \sqrt{y_j} \right)^2}$	$[0, \infty)$	$\bar{d} = \frac{d_{XY}}{\sqrt{n}}$	$[0, \infty)$
Wave–Edges (m15)	$d_{XY} = \sum_{j=1}^{h} \left(1 - \frac{\min \left\{ x_j, y_j \right\}}{\max \left\{ x_j, y_j \right\}} \right)$	$[0, n]$	$\bar{d} = \frac{d_{XY}}{n}$	$[0, 1]$
Angular Separation/[1-Cosine (Ochiai)] (m16)	$d_{XY} = 1 - \mathrm{Cos}_{XY}$ where, $\mathrm{Cos}_{XY} = \frac{\boldsymbol{XY}}{\lVert \boldsymbol{X} \rVert \lVert \boldsymbol{Y} \rVert} = \frac{\sum_{j=1}^{h} x_j y_j}{\sqrt{\sum_{j=1}^{h} x_j^2 \sum_{j=1}^{h} y_j^2}}$	$[0, 2]$		

[a] The variable $x_j(y_j)$ is the value of the coordinate j of the atom s and the atom t, corresponding to the molecule X (Y), respectively. The h value is the Cartesian coordinates (x, y, z) of an atom. The p values in Minkowski metric are 0.25, 0.5, 1 (Manhattan), 1.5, 2 (Euclidean), 2.5 and 3 (Minkowsky). [b] “Range” refers to “range” and not to “rank” and is defined as Range $= \max \left\{ x_j \right\} - \min \left\{ x_j \right\}$.

atom-types belonging to a specific molecular fragment (F). To this end, the kth local-fragment two-tuples, three-tuples, and four-tuples spatial-(dis)similarity matrices, $\mathbb{G}_F^k$, $\mathbb{GT}_F^k$, and $\mathbb{GQ}_F^k$ are utilized, respectively.[10,11] From these total local-fragment matrices, the corresponding atom-level local-fragment matrices ($\mathbb{G}_F^{a,k}$, $\mathbb{GT}_F^{a,k}$, $\mathbb{GQ}_F^{a,k}$) are determined [see Section 4 in Supporting Information 1], which are used to compute the atom-level local-fragment indices [see eqs. (1)–(3)]. The molecular fragments (or atom-types) used are: hydrogen bond acceptors (A), carbon atoms in aliphatic chains (C), hydrogen bond donors (D), halogens (G), terminal methyl groups (M), carbon atoms in aromatic portion (P), and heteroatoms [O (oxygen), N (nitrogen), and S (sulfur) in all valence states, denoted as X].

Moreover, with the aim of taking into consideration only some type of interatomic interactions (e.g., short-, middle-, and large-contacts) in the computation of the QuBiLS-MIDAS MDs, two different constraints have been used: (1) N-tuples Graph-theoretical cutoff (p) based on topological distance at a lag p, denoted as “path cutoff” and (2) N-tuples Geometric cutoff (l), based on Euclidean distance at a lag l known as “length cutoff.” With the application of these cutoffs over the matrix $\mathbb{G}^1$ [$\mathbb{GT}^1$, $\mathbb{GQ}^1$], a new matrix approach is created: the two-tuples [three-tuples, four-tuples] topological and geometric neighborhood quotient matrix, $\mathbb{NQG}^1$ [$\mathbb{NQGT}^1$, $\mathbb{NQGQ}^1$]. These two cutoffs permit to unify the topological (2D) and geometric (3D) information in a same matrix representation, as well as, to take into account the most important noncovalent interactions according to user-predefined thresholds p and/or l.

Finally, when normalizing procedures are not employed for the previous matrix approaches, then these are known as the kth non-stochastic [two-tuples, three-tuples, and four-tuples] total (or local-fragment) spatial-(dis)similarity matrices [${}_{ns}\mathbb{G}_{(F)}^k$, ${}_{ns}\mathbb{G}_{(F)}^k$, and ${}_{ns}\mathbb{GQ}_{(F)}^k$]. However, the probabilistic transformations have been used in other frameworks with successful results.[16,18,45–47] So, with the purpose of normalizing the nonstochastic N-tuples matrices three probability schemes are applied: the simple-stochastic, the double-stochastic, and the mutual probability algebraic transformation. The double-stochastic scaling is only applied to the two-tuples matrices[48,49] due to the fact that no similar procedure has been reported for high-order matrices (see Section 3 in Supporting Information 1). Specifically, the kth simple-stochastic spatial-(dis)similarity matrices [${}_{ss}\mathbb{G}_{(F)}^k$, ${}_{ss}\mathbb{GT}_{(F)}^k$, and ${}_{ss}\mathbb{GQ}_{(F)}^k$] are nonsymmetric, where the sum of the elements of each row is equal to 1. Conversely, in the kth double-stochastic spatial-(dis)-similarity matrices [${}_{ds}\mathbb{G}_{(F)}^k$], the sum of the elements of each row and column is approximately equal to 1 and lastly, in the kth mutual-probability spatial-(dis)similarity matrices [${}_{mp}\mathbb{G}_{(F)}^k$, ${}_{mp}\mathbb{GT}_{(F)}^k$, and ${}_{mp}\mathbb{GQ}_{(F)}^k$] the sum of all coefficients is equal to 1.

Overview of the QUBILS-MIDAS Software

To compute the 3D-MDs previously described, the QuBiLS-MIDAS software was developed. The implementation of this computer program was performed in Java (version 1.7) due to the advantage presented by this programming language in being fully cross-platform. The programs written in this language are compiled to byte-code format, which permit the running of applications in any operating system and hardware whenever there is a Java Virtual Machine (JVM) available. It is important to highlight, that in the development of this software the Chemical Development Kit library[50] (version 1.4.19) was employed, mainly for the manipulation and storing of the molecular structures. Also, it was used for the calculation of the atom- and fragment-based chemical properties utilized in the QuBiLS-MIDAS program as weighting schemes for the atoms of a molecule.

The QuBiLS-MIDAS software consists of two main components: the front-end and the back-end. In the front-end, the graphical user

Table 3. Measures used to compute the ternary and quaternary relations for atoms of a molecule.

Measure	Formula
Ternary Measures (TT_{XYZ})	
Perimeter (m19–m20)	$T_{XYZ}=d_{xy}+d_{yz}+d_{zx}$
Triangle Area (m21–m22)	$T_{XYZ}=\sqrt{s(s-d_{XY})(s-d_{YZ})(s-d_{ZX})}$ $s=\frac{d_{XY}+d_{YZ}+d_{ZX}}{2}$
Summation Sides (m25–m26)	$T_{XYZ}=d_{XY}+d_{YZ}$
Bond angle (Angle between sides) (m27–m28)	$A_X,\ A_Y,\ A_Z$ coordinates of three atoms of a molecule $U=A_X-A_Y,\ V=A_Z-A_Y$ $T_{XYZ}=\alpha=\arccos\left(\frac{U\times V}{\lvert U\rvert\times\lvert V\rvert}\right)$
Quaternary Measures (QQ^{XYZW})	
Perimeter (m19–m20)	$Q_{XYZW}=d_{XY}+d_{YZ}+d_{ZW}+d_{WX}$
Volume (m23–m24)	$A_X,\ A_Y,\ A_Z,\ A_W$ coordinates of four atoms of a molecule $Q_{XYZW}=\frac{1}{6}\begin{pmatrix} A_{Y1}-A_{X1} & A_{Z1}-A_{X1} & A_{W1}-A_{X1} \\ A_{Y2}-A_{X2} & A_{Z2}-A_{X2} & A_{W2}-A_{X2} \\ A_{Y3}-A_{X3} & A_{Z3}-A_{X3} & A_{W3}-A_{X3} \end{pmatrix}$
Summation Sides (m25–m26)	$Q_{XYZW}=d_{XY}+d_{YZ}+d_{ZW}$
Dihedral Angle (m29–m30)	$A_X,\ A_Y,\ A_Z$ coordinates of three atoms of a molecule in the plane A $B_W,\ B_Y,\ B_Z$ coordinates of three atoms of a molecule in the plane B $U_A=(A_X-A_Y)\times(A_Z-A_y)$ $U_B=(B_W-A_Y)\times(B_Z-A_y)$ $Q_{XYZW}=\alpha=\arccos\left(\frac{U_A\times U_B}{\lvert U_A\rvert\times\lvert U_B\rvert}\right)$

interfaces (GUIs) are implemented for the configuration of the MDs, whereas in the back-end the responsible classes of performing the computation of these molecular parameters are defined. The back-end was developed as an Application Programming Interface (API), in such way that it can be used as Java library in the development of other software or in the implementation of other user-friendly interfaces either graphically or command-line based. In this way, with these two components it is achieved that the presentation layer of the QuBiLS-MIDAS software is independent of the processing logic implemented in the back-end and thus, any modification in this last component does not provoke changes in the front-end, and vice versa. Table 4 shows a comparison for several MD calculating software, with the respective features detailed.

Front-end: Desktop graphic user interface of the QuBiLS-MIDAS software

To facilitate the calculation of the QuBiLS-MIDAS MDs, a remarkably friendly Desktop GUI was developed to provide a simple way to configure the different parameters used, such as: the algebraic forms, the matrix approaches, the *N*-tuples cutoffs, atomic properties, and so on. Figures 1 and 2 show the principal user interface of the program (GUI) and the dialog windows designed to configure some of these parameters, respectively. These configuration sections constitute distinctive features of this software with respect to remaining applications, because they allow to the users to personalize the *N*-linear indices according to their necessities and thus predefined MDs are not calculated, for instance: the DRAGON software computes the radial distribution function (RDF) MDs represented by the nomenclatures RDFRw, where the *R* values (radius of the spherical volume) and the atomic properties *w* are established by default and not by the user.

Therefore, in the QuBiLS-MIDAS program, in the "Algebraic Form" panel one may chose the specific algebraic maps to be used in the computation of the MDs according to the "*N*" atoms to be taken into account, which are selected in the "Constraints" panel as follows: the Duplex option for $N=2$,three-tuples (ternary) option for $N=3$ and four-tuples (quaternary) option for $N=4$. The matrix approaches

Table 4. Main features and qualitative general comparison of 12 commonly used tools for molecular descriptors (MDs) calculations.

Software	Number of types of MDs	Configuration of MDs parameters	Advantages	Disadvantages	Additional remarks
QuBiLS-MIDAS v1.0	2080 (bilinear, quadratic and linear) 5520 (three-linear) 13440 (four-linear)	1. Atomic properties 2. (Dis-)Similarity metrics and multimetrics 3. Local-fragments 4. Matrix approaches 5. Aggregation operators 6. Add (or remove) hydrogen atoms 7. Consider lone-pair electrons or distance of each atom to molecule center	1. Computes MDs based on relations among three and four atoms 2. Graphic user-friendly interface and command-line interface 3. Fully cross-platform 4. Supports any organic molecules 5. Free download 6. Batch mode processing 7. Data cleaning module 8. Parallel processing 9. 10 atom weighting schemes	1. Only accepts MDL files (MOL or SDF) as input formats	1. Uses CDK to read molecular files and calculate atomic properties 2. Requires Java JRE 1.7 or above
PaDEL-Descriptor v2.0	43	None	1. Graphic user interface 2. Fully cross-platform 3. Command line interface 4. Free and Open Source 5. Accepts multiple file formats (>90 formats) 6. Parallel processing	1. One functionality for data cleaning tasks (remove salts) 2. No MDs batch processing	1. Uses CDK to read molecular files and calculate most of the descriptors and fingerprints. 2. Employs Java Web Start technology
DRAGON v6.0	29	1. Choose the atom weighting schemes 2. If apply logarithmic transformation to spectral moments 3. Selection of single molecular descriptors included in the different blocks	1. Graphic user-friendly interface 2. Command line interface 3. Batch mode processing 4. Supports any organic molecules 5. Accepts the formats: MDL, Sybyl, HyperChem, Macromodel, Smiles, CML and HyperChem	1. Only Windows and Linux platforms 2. No parallel processing 3. No data cleaning functionalities 4. Does not allow selection of local-fragment 5. Commercial cost	Academic permanent license: 900 euro (to be installed on 3 PCs) http://www.talete.mi.it/products/dragon_description.htm
CDK Descriptor Calculator v1.3.9	48	1. Add (or remove) hydrogen atom	1. Graphic user interface 2. Command line execution, Free 3. Fully cross-platform 4. Free software 5. Batch mode processing	1. Only accepts MDL files (MOL or SDF) as input formats 2. No data cleaning functionalities 3. Does not allow selection of local-fragment 4. Does not allow selection of atom weighting schemes	Use CDK library and requires Java JRE 1.6 http://www.rguha.net/code/java/cdkdesc.html

(Continued)

Table 4. (*Continued*)

Software	Number of types of MDs	Configuration of MDs parameters	Advantages	Disadvantages	Additional remarks
BlueDesc	36	None	1. Free and Open Source 2. Fully cross-platform	1. No graphic user interface 2. Only accepts MDL files (MOL or SDF) as input formats 3. No parallel processing 4. No data cleaning functionalities 5. Does not allow selection of local-fragment 6. Does not allow selection of atom weighting schemes	Use CDK and JOELib2 library and requires Java JRE 1.6 http://www.ra.cs.uni-tuebingen.de/software/bluedesc /welcome_e.html
Model	98	None	1. Web-based graphic user interface 2. Accepts the formats: PDB, MDL, MOL2,COR	1. No parallel processing 2. No data cleaning tasks 3. Does not allow selection of local-fragment 4. Does not allow selection of atom weighting schemes 5. For academic purposes only	http://jing.cz3.nus.edu.sg/cgi-bin/model/model.cgi
Mol2	20	None	1. Command line interface 2. Free of charge download request	1. No graphic user interface 2. Only Windows platform 3. Only accepts SDfile format 4. No parallel processing 5. No data cleaning functionalities 6. Does not allow selection of local-fragment 7. Does not allow selection of atom weighting schemes	http://www.fda.gov/ScienceResearch /BioinformaticsTools/Mold2/ucm144528.htm
MOE	–	None	1. Graphic user interface 2. Command line interface 3. Data cleaning tasks 4. Fully cross-platform	1. Only accepts SDfile format 2. No parallel processing 3. Does not allow selection of local-fragment 4. Does not allow selection of atom weighting schemes	http://www.chemcomp.com/MOE-Cheminformatics_and_QSAR.htm

(Continued)

Table 4. (*Continued*)

Software	Number of types of MDs	Configuration of MDs parameters	Advantages	Disadvantages	Additional remarks
VolSurf	22	None	1. Graphic user interface 2. Command line interface 3. Accepts several formats: MDL SDF, Sybyl, Mol2, Multi Mol2, GRID kout.	1. Commercial 2. Only Linux platform 3. Only compute 2D molecular descriptors 4. No parallel processing 5. Does not allow selection of local-fragment 6. Does not allow selection of atom weighting schemes	http://www.moldiscovery.com/soft_volsurf.php
Adriana.C	5	None	1. Graphic user interface 2. Command line interface 3. Batch mode processing 4. Accepts any organic molecule	1. Commercial 2. Only Windows and Linux platforms 3. No parallel processing 4. No data cleaning functionalities 5. Does not allow selection of local-fragment 6. Does not allow selection of atom weighting schemes	http://www.molecular-networks.com/products/adrianacode
CODESSA PRO	8	None	1. Graphic user interface	1. Commercial 2. Only for Windows platform 3. No parallel processing 4. No batch mode processing 5. Does not allow selection of local-fragment 6. Does not allow selection of atom weighting schemes	http://www.codessa-pro.com/
PowerMV	–	None	1. Graphic user interface	1. Only for Windows platform 2. No parallel processing 3. No batch mode processing 4. Does not allow selection of local-fragment 5. Does not allow selection of atom weighting schemes	Requires Microsoft .net 1.1 or above http://nisla05.niss.org/PowerMV

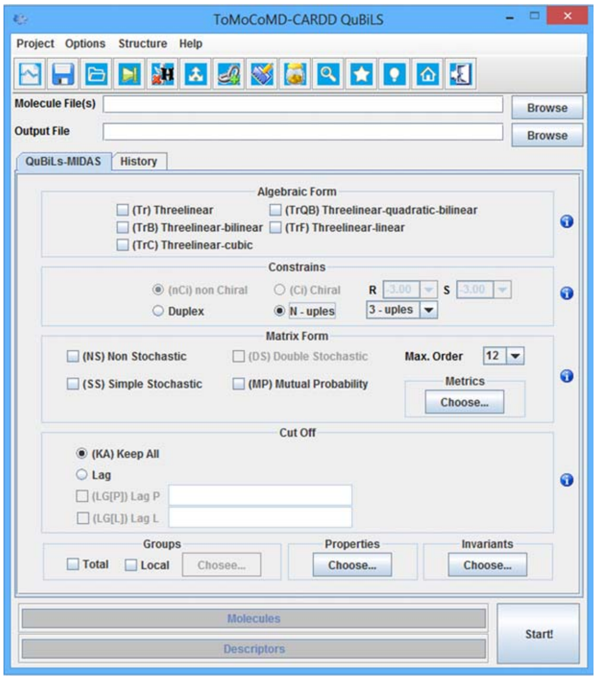


Figure 1. Principal graphic user interface for the QuBiLS-MIDAS software.

(nonstochastic, simple-stochastic, double-stochastic, and mutual probability) used in the algebraic forms are configured in the "Matrix Form" panel. In this same panel, the (dis-)similarity metrics and multimetrics to characterize the relations between "*N*" atoms of a molecule are also chosen, as well as, the power *k* (order) to which the coefficients of the matrices are raised. In the "CutOff" panel, one may select the option to analyze the whole molecule or specify the value(s) and/or value-rank(s) of the thresholds *p* and/or *l* to consider only the noncovalent interactions under the established criterion. Lastly, in the "Local-Fragments" panel, the chemical groups (or atom-types) to compute the local-fragment *N*-linear indices are selected; in the "Properties" panel, the atomic properties used to set different weighting schemes are chosen as components of the molecular vectors and accordingly, several combinations of the *N*-linear forms are achieved (see Table 1); and in the "Invariants" panel, the mathematical operators used to aggregate the atomic contributions are configured in order to obtain different total or local MDs.

Moreover, in the GUI there are other useful options to configure the QuBiLS-MIDAS MDs. In this way, the "On/Off H-Atoms" option could be used to consider or not the hydrogen (H) atoms during the calculation of the indices (by default the H-depleted structures are considered). Besides, nonzero values in the diagonal coefficients may be taken into account by selecting the "On/Off Distance to Center" option to compute the diagonal coefficients as the distance of each atom to center of the molecule, or the "On/Off Lone-Pair Electron" option to consider the amount of lone-pair electrons present in each atom. It is important to point out that these last two options cannot be chosen at the same time. Also, the QuBiLS-MIDAS program has the "Show Debug Report" option, which can be utilized to save all information concerning to the algebraic process that takes place in the calculation.

In addition, this application permits to choose the input files corresponding to the chemical structures to be analyzed,

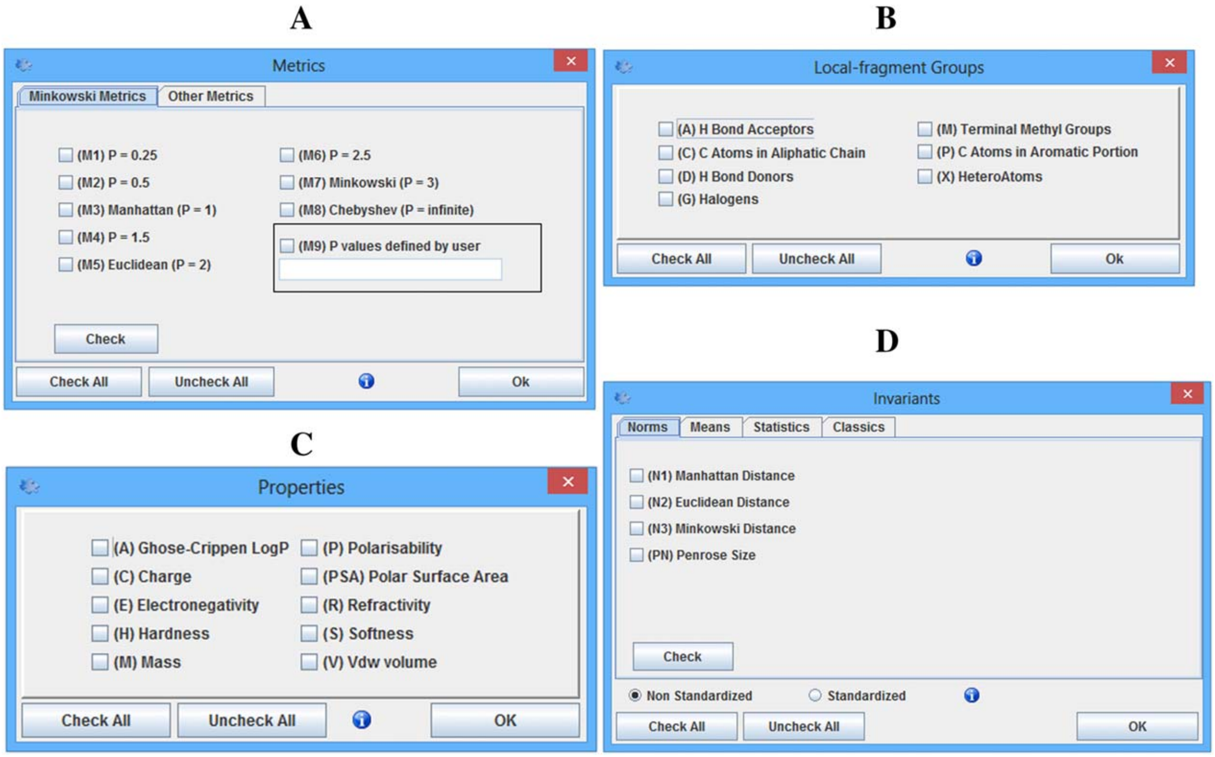
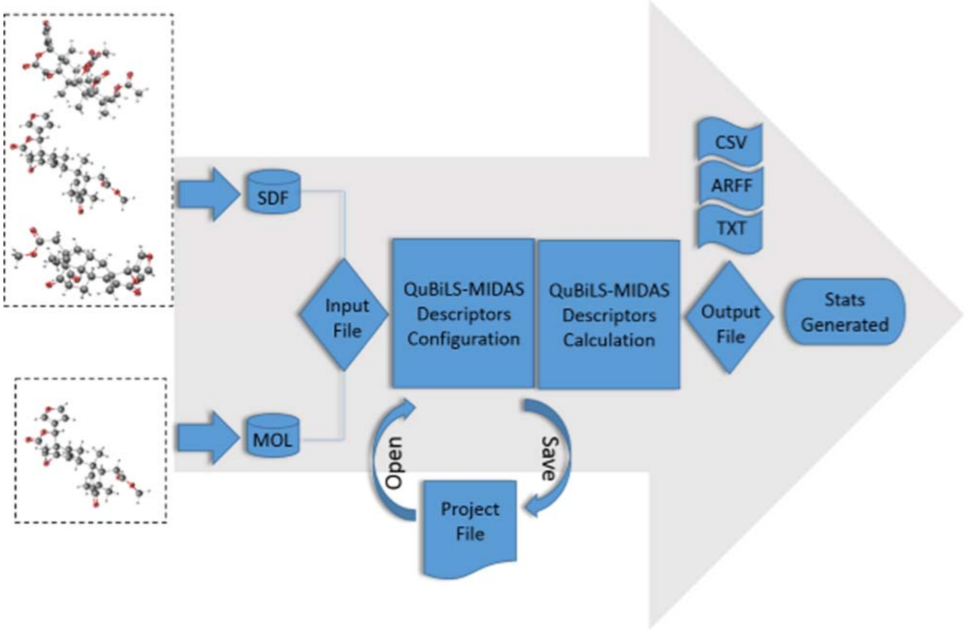


Figure 2. Dialog windows to configure some of the parameters to compute the QuBiLS-MIDAS indices. (A) Interface for the (dis-)similarity metrics. (B) Interfacefor the local-fragment (atom-type) chemical groups. (C) Interface for the atom properties. (D) Interface for invariants or aggregation operators.


SDF
MOL
Input File
QuBiLS-MIDAS Descriptors Configuration
QuBiLS-MIDAS Descriptors Calculation
CSV
ARFF
TXT
Output File
Stats Generated
Open
Save
Project File

Scheme 1. General workflow of the QuBiLS-MIDAS software for descriptors calculating. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

which may be supported in the MDL MOL/SDF formats. These input files are sequentially read due to the fact that only one molecular structure is analyzed at a time and in this manner, the program employs suitable memory allocation according to the size of the molecule and to the matrix form to be used. Moreover, the path of the output file may be specified where the values of the computed MDs are saved. To this end, the QuBiLS-MIDAS program offers the following output file formats: CSV, ARFF, and TXT (with the space ASCII separated format) which are easily interpretable by popular statistical and/or machine learning applications, such as MobyDigs[51] and WEKA[52] (see Scheme 1 for program workflow).

Another feature of this program is that it enables saving the configuration utilized for the calculation of MDs to a XML file (projects). In this manner, several configurations can be saved as QuBiLS-MIDAS projects, which may subsequently be used with other molecular datasets when the software is run again. This option is important due to the fact that it is not necessary to manually configure the same MDs in the GUI each time they need to be computed. Furthermore, in this software several projects are provided by default for the calculation of the QuBiLS-MIDAS Duplex, Ternary, and Quaternary MDs, which can be used by the final-user to perform chemoinformatics studies. These projects have been employed successfully (better outcomes than those reported in the literature) in the modeling of the binding affinity to the corticosteroid-binding globulin, and were built from the best results obtained in in-house comparisons according to relevance, orthogonality, and correlation ability (QSAR study) of the different parameters used for the computation of the QuBiLS-MIDAS 3D-indices.[10,11]

The calculation of the MDs can be performed in an interactive mode using the GUI and its progress monitored through the main interface. This procedure has an inconvenience in that if the user wants to execute several QuBiLS-MIDAS projects, then these have to be sequentially run (i.e., one by one), which constitute a tedious task. However, this software has a batch processing module implemented which allows an easy integration for high-throughput and routinely carried out MD calculations. In reality, this module is designed to manage and allow the configuration of up to eight batch tasks, where one task is constituted by one or several datasets on which one or several projects previously saved are computed.

Finally, it is important to remark that the QuBiLS-MIDAS software has incorporated a module for data curation tasks. In Ref. [53], Tropsha and coworkers, motivated by the errors existing in some medicinal chemistry publications due to the inaccuracy of the input data,[54] investigates several errors present in popular public databases of compounds which have affected the results of chemoinformatics studies. For this reason, these authors propose a guideline with the most common curation practices that may be followed as a general reference scheme. In this sense, the QuBiLS-MIDAS software has incorporated functionalities for the removal of mixtures, inorganics, organometallics, and duplicates, as well as, the neutralization of salts. These procedures were implemented according to the description performed in Tropsha's guidelines.[53]

Back-end: The core classes to compute the QuBiLS-MIDAS MDs

All the requests performed by the users through the GUI are processed by the QuBiLS-MIDAS library. This component is organized in packages according to the goals of the functionalities and so to facilitate its understanding. The principal package is tomocomd.cardd.qubils, which does not contain

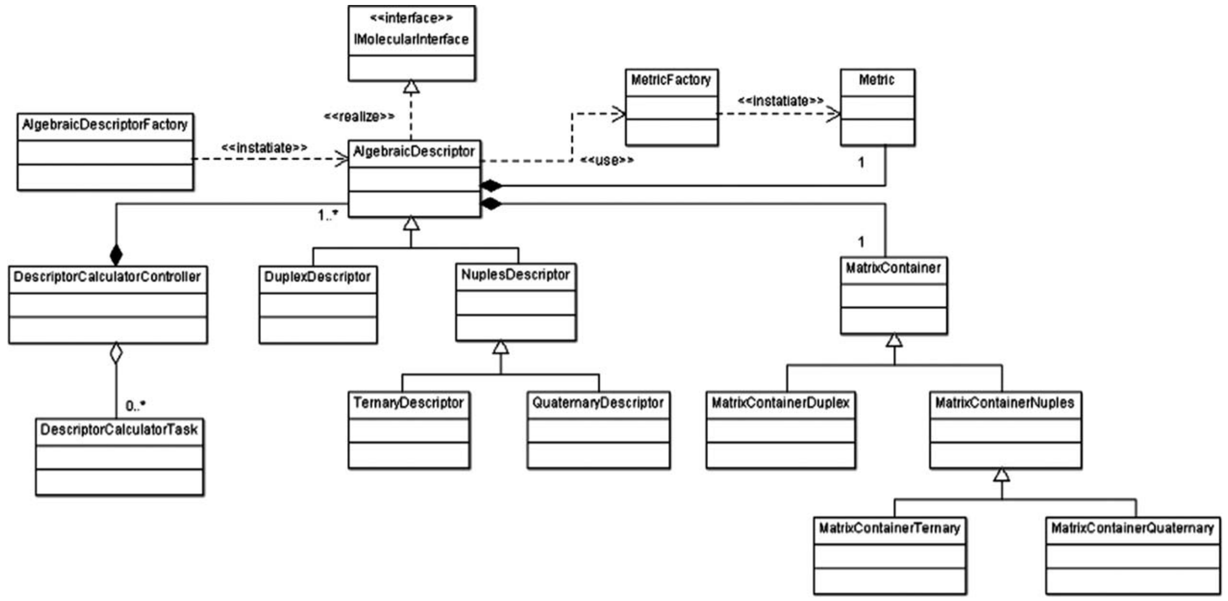


Figure 3. UML Class Diagram for the most important classes that contribute to the calculation of the QuBiLS-MIDAS indices.

objects' definitions, but contains the packages descriptors, matrices, metrics, and workers that encapsulate the main concepts utilized in the definition of the QuBiLS-MIDAS MDs and in the implementation of this program.

The descriptors package includes the classes related with the bilinear, quadratic, and *n*-linear algebraic maps. The matrices package contains the objects responsible for building the *N*-tuples matrix ($2 \leq N \leq 4$) representations that are used by the algebraic forms. The metrics package presents the classes corresponding to the (dis-)similarity metrics based on the relations for two, three, and four atoms of a molecule. Finally, the workers package has the necessary classes for the configuration, realization and control of the calculation of the MDs implemented.

Figure 3 shows the UML diagram belonging to most important classes implemented to perform the computation of the QuBiLS-MIDAS MDs, while Figure 4 shows the UML diagram corresponding to the functions employed to compute the relations for *N* atoms. As can be observed, the classes AlgebraicDescriptor, Metric, and MatrixContainer constitute the superclasses that support abstraction levels for the concepts related with the *N*-linear molecular indices, the (dis-)similarity metrics, and the *N*-tuples matrices, respectively.

In Figure 3, the classes DuplexDescriptor, TernaryDescriptor, and QuaternaryDescriptor have defined the concepts corresponding to the MDs based on two-linear (bilinear, quadratic, and linear), three-linear, and four-linear algebraic maps, respectively. Conversely, the classes MatrixContainerDuplex, MatrixContainerTernary, and MatrixContainerQuaternary represent the two-, three-, and four-tuples spatial-(dis)similarity matrices. In addition, these classes contain the algorithms to perform

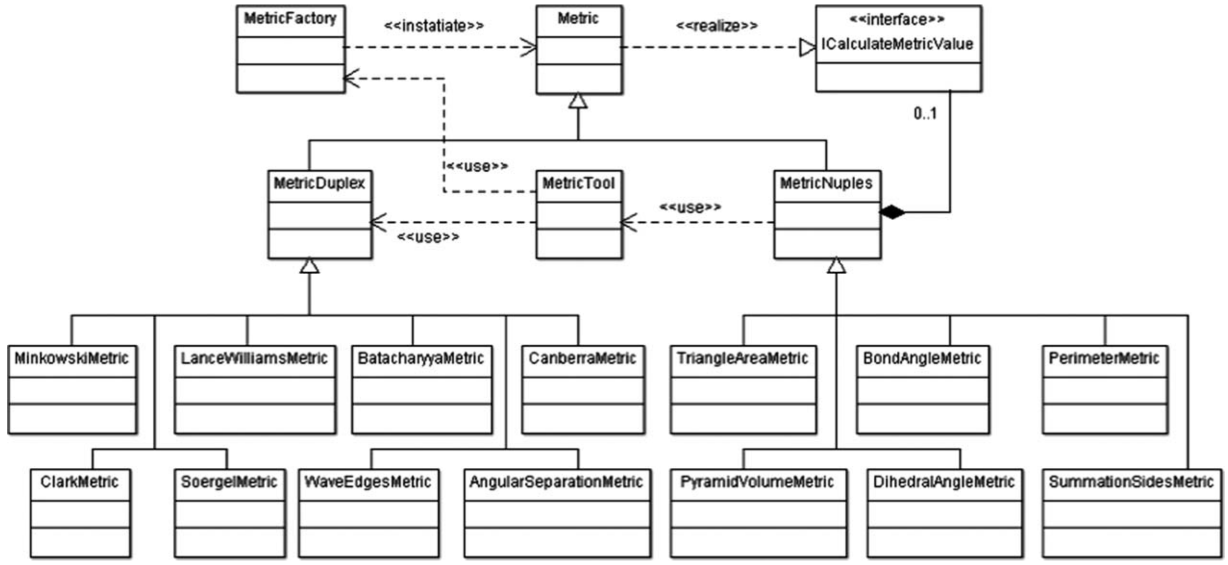


Figure 4. UML Class Diagram for the responsible classes to compute the (dis-)similarity metrics based on the relations for two, three, and four atoms of a molecule.

the simple-stochastic, double-stochastic and mutual probability algebraic transformations, as well as, to determine the atom-level spatial-(dis)similarity matrices. In Figure 4, the classes that extend from MetricDuplex have the distance measures (Minkowski, Canberra, Soergel, and so on) implemented so as to establish relations between two atoms, whereas the classes that extend from MetricNuples represent the metrics to compute relations among three and four atoms.

Lastly, the classes AlgebraicDescriptorFactory and MetricFactory handle the creation of the different kinds of descriptors and (dis-)similarity metrics either from a configuration performed by the user through the GUI or from a QuBiLS-MIDAS project previously saved. Finally, the DescriptorCalculatorController class is responsible for controlling the computation of the MDs and to store the results obtained in an output file according to the format selected by the user; whereas the DescriptorCalculatorTask class is accountable for performing the calculation.

The processing time of this program relies on the high-throughput computing of *N*-linear indices and/or molecular datasets, for instance: if the sequential calculation of one four-linear index has a computational complexity of $O(n^4)$ [*n* is the dimension of the algorithm, in this case constitutes the amount of atoms of the molecule] and 5000 MDs are computed over a dataset of 10,000 compounds (considering all compounds to contain 50 atoms each), then the program performs 3125×10^{11} floating-points operations. Thus, this calculation accomplished in a processor capable of executing 5×10^9 operations per second will delay approximately $62,500\ \mathrm{seg}$. If this same computation is performed in four processors with equal computational power, then the processing is reduced to approximately $15,600\ \mathrm{seg}$.

Bearing this in mind, the algorithms for calculating the QuBiLS-MIDAS MDs were efficiently implemented. Furthermore, taking into consideration the parallel environmental of the modern computers and the evident decomposition of the calculation to be performed into independent-tasks, the presented software was developed to take advantage of the multicore architectures and thus to improve the speed in the calculation. These previous ideas constitute essential features for the development of software for the calculation of MDs, which for this program are explained in the next section.

Analysis of Algorithms and Parallel Performance Tests

In this section, the complexity of the main algorithms implemented for the calculation of the QuBiLS-MIDAS 3D-indices is analyzed. For each algorithm the corresponding asymptotic notation (Big O) is shown. Also, the parallelization strategy and the tests used to analyze the speed achieved in the computation of the MDs are detailed.

Temporal complexity of algorithms

The analyzed algorithms are the corresponding to simple-stochastic and mutual-probability algebraic transformations, as well as, the procedure to compute the atomic contributions of the QuBiLS-MIDAS Duplex, Ternary, and Quaternary MDs. The analysis of the algorithm for the double-stochastic transformation is performed in reference,[49] where the Sinkhorn–Knopp method is presented.

The simple-stochastic scaling, in general sense, is performed by dividing each entry of the matrix by the sum of all coefficients corresponding to the slide to which the entry belongs. In this way, the relations of each atom *i* with respect to the remaining atoms of the molecule are taken into account. Specifically, this procedure in a two-tuples matrix is carried out by dividing each coefficient of a row (slide) by the total sum of the entries belonging to the same row (see eq. (14) in Supporting Information 1); in a three-tuples matrix this is attained through the division of each entry by the summation of all entries of the two-tuple matrix (slide) corresponding to each atom *i* (see eq. (15) in Supporting Information 1); and in a four-tuples matrix each coefficient is divided by the total sum of all entries of the three-tuple matrix (slide) belonging to each atom *i* (see eq. (16) in Supporting Information 1). Thus, the computational cost of this algorithm for the two-, three-, and four-tuples matrices is $O(n^2)$, $O(n^3)$, and $O(n^4)$, respectively.

Conversely, the mutual-probability transformation, irrespective of the matrix order, is accomplished by summing all coefficients of the corresponding matrix and next, each coefficient of the matrix is divided by the calculated total sum (see eqs. (17)–(19) in Supporting Information 1). Therefore, this procedure applied to the nonstochastic two-, three-, and four-tuples matrices [$_{ns}\mathbb{G}^k_{(F)}$, $_{ns}\mathbb{GT}^k_{(F)}$, and $_{ns}\mathbb{GQ}^k_{(F)}$] present a computational complexity of $O(n^2)$, $O(n^3)$, and $O(n^4)$, respectively.

Finally, the algorithms for the calculation of the QuBiLS-MIDAS MDs are analyzed. These constitute the principal procedures due to the fact that they are the responsible for performing the multiplication based on two-linear (bilinear, quadratic, linear), three-linear, and four-linear algebraic forms to compute the atom-level indices.

The trivial algorithm to compute the atomic contributions consists in the use of a loop that iterates for each atom of a molecule and in this way, determines the corresponding atom-level matrix. Inside this loop, the strategy corresponding to the iterations for the two-, three-, or four-tuples matrices is implemented to analyze the relations for two, three, or four atoms with respect to the atom represented by the outer loop. Once the atom-level matrices are determined, then these are multiplied by the corresponding property vectors and subsequently, the atom-level indices are obtained. These implementations have a computational complexity of $O(n^3)$, $O(n^4)$, and $O(n^5)$ when MDs based on relations for two, three, and four atoms are calculated, respectively.

However, in the previous procedures when the atom-level matrices are computed, it can be analyzed that the only entries with values different from zero are the corresponding to the atom with respect to which the atom-level matrix is built. Therefore, this suggests that in place of iterations for each atom to build the atom-level matrix and then determine the corresponding atom-level index, it is more efficient to compute the atom-level indices at the same time that the two-, three-, or four-tuples matrices are analyzed. In this manner, these

Table 5. Calculation results of the QuBiLS-MIDAS indices.

Number of processors	Processing time	Speedup	Efficiency	Processing time for one molecule (s)	Processing time for one descriptor (s)
QuBiLS-MIDAS duplex indices					
1	2139	1.000	1.000	2.139	0.105
2	1382	1.547	0.774	1.382	0.068
4	932	2.295	0.574	0.932	0.046
8	586	3.653	0.457	0.586	0.029
16	383	5.584	0.349	0.383	0.019
QuBiLS-MIDAS ternary indices					
1	31,847	1.000	1.000	31.847	1.570
2	16,306	1.953	0.977	16.306	0.804
4	9079	3.508	0.877	9.079	0.448
8	6528	4.879	0.610	6.528	0.322
16	4633	6.874	0.430	4.633	0.228
QuBiLS-MIDAS quaternary indices					
1	639,006	1.000	1.000	639.006	31.509
2	379,391	1.684	0.842	379.391	18.708
4	213,830	2.988	0.747	213.830	10.544
8	152,421	4.192	0.524	152.421	7.516
16	106,370	6.007	0.375	106.370	5.245

algorithms have been optimized to an inferior polynomial order, with a complexity of $O(n^2)$, $O(n^3)$, and $O(n^4)$ for the computation of the atomic contributions corresponding to the QuBiLS-MIDAS Duplex, Ternary, and Quaternary MDs, respectively.

Speed-up analysis in the parallel processing

The QuBiLS-MIDAS MDs are computed in parallel using the Fork/Join framework[55] implemented in Java. So, there is a pool with many Worker threads according to the available CPUs on the system, where each worker has its own scheduling queue. A recursive task (DescriptorCalculatorTask object) that contains the total set of MDs to compute is responsible for dividing this calculation into smaller tasks (DescriptorCalculatorTask objects) depending on the amount of workers. When these subtasks only contain one descriptor then it is directly calculated, or else these subtasks continue being divided until the previous threshold is achieved. It is important to highlight, that in this framework each subtask created by the task of a given worker is pushed to its queue. In addition, when a worker does not have more tasks that attend to then it takes tasks from other workers' queue randomly selected. The advantage of the Fork/Join framework is that it makes an efficient use of all CPUs present in modern computers through the utilization of the work-stealing schedule strategy.[56]

The experiments to analyze the speed achieved during the calculation of the QuBiLS-MIDAS MDs were performed on a Dell Precision T7600 computer workstation with two Intel® Xeon® E5–2687W 3.1 GHz processors and 32 GB RAM, but only 16 GB RAM were assigned to the JVM to run the QuBiLS-MIDAS software. This architecture integrates eight native cores with simultaneous multithreading technology each one, which enables the execution of 16 processing threads at the same time. A total of 20,280 QuBiLS-MIDAS Duplex, Ternary, and Quaternary MDs were computed for the PrimScreen1 collection database (http://www.otavachemicals.com/download-compound-libraries/cat_view/110-diversity-sets/128-primscreen-1) comprised of 1000 structures.

Table 5 shows the processing time (total, per molecule, and descriptor), speedup, and efficiency achieved in the calculation of the QuBiLS-MIDAS Duplex, Ternary, and Quaternary MDs. Also, in Figure 5 the processing time and speedup, corresponding to QuBiLS-MIDAS indices, are graphically represented. As can be observed, the total processing time always decreases as long as the number of processors is increased (see Figs. 5A–5C), and thus the multicore architecture is properly employed. In this way, the sequential calculation time of the Duplex, Ternary, and Quaternary indices is approximately reduced in 5.8, 6.8, and 6.0 times, respectively.

However, as it can be seen in Figure 5D, the speedup is not always proportional to the amount of processors used. Consequently, it can be appreciated in the Table 5 (Efficiency column) that the efficiency considerably decreases beyond four processors in the computation of the QuBiLS-MIDAS indices. This behavior could be motivated by the fact that there exists a greater level of synchronization among the different worker threads when the descriptor values are sent to the output file, due to the fact that this process must be performed in the same order of creation of the molecular indices.

Conclusions

Free software for the computation of the QuBiLS-MIDAS MDs, composed of a desktop user-friendly interface (GUI) and a library component (API) was developed. Therefore, it can be used as standalone software or as part of another application. This software was implemented in Java and thus can be executed in different architectures and operating systems. In addition, it allows configuration of each parameter of the *N*-linear indices, which constitutes a unique feature with respect to the remaining calculation programs of MDs. Also, this software has incorporated a set of functionalities for data curation tasks and a module for the batch processing of MDs. Lastly, for the computation the

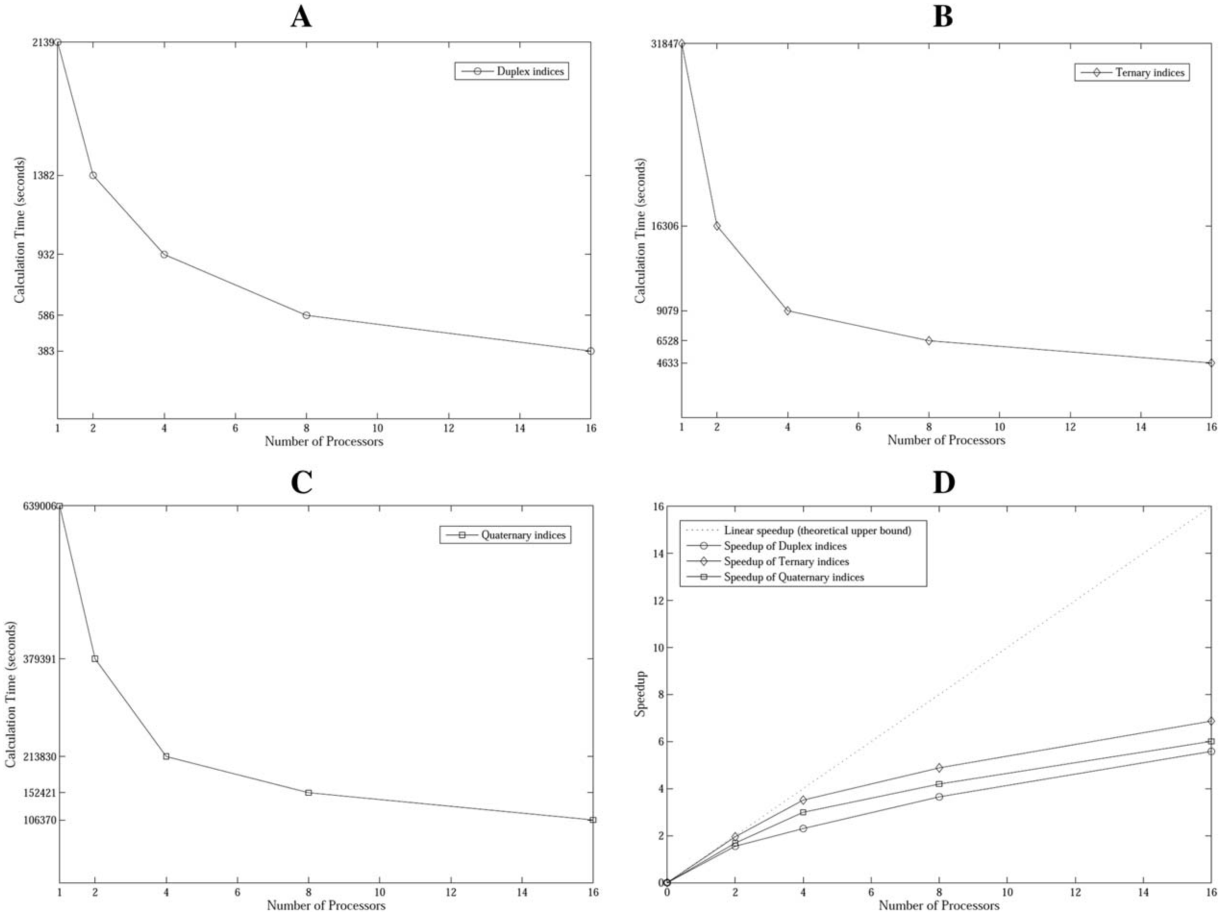


Figure 5. Charts of processing time and speedup in the calculation of the QuBiLS-MIDAS indices. (A) Execution time of the QuBiLS-MIDAS duplex indices. (B) Execution time of the QuBiLS-MIDAS ternary indices. (C) Execution time of the QuBiLS-MIDAS quaternary indices. (D) Speedup achieved during the calculation of the QuBiLS-MIDAS duplex, ternary and quaternary indices.

multicore architecture of the current computers is utilized and so the speed during the processing is increased.

Futures Outlooks

The QuBiLS-MIDAS software is being continuously updated. At the moment, a second version is being developed where chirality properties, more (dis-)similarity metrics and multimetrics, more local-fragments or atomic properties are taken into consideration. Furthermore, a new set of cutoffs based on multimetrics and spherical truncates are being implemented, as well as, a module for the reduction of nonrelevant descriptors according to a determined criterion and another module for structure visualization. Lastly, a version to compute molecular indices on a distributed computing system for high-throughput calculation tasks is being developed, as well as, a version to use the Graphical Processing Unit (GPU) present in several personal computers nowadays.

Acknowledgments

Yovani Marrero-Ponce thanks the program "International Professor" for a fellowship to work at Cartagena University in 2013–2014. Program availability: The QuBiLS-MIDAS software and the corresponding user's manual are freely available in the ToMoCoMD Framework web site (www.tomocomd.com). For obtaining the source code of the program, please to write to the corresponding author: ymarrero77@yahoo.es or ymponce@gmail.com.

Keywords: TOMOCOMD-CARDD · QuBiLS-MIDAS · 3D *N*-linear indices · *N*-tuple spatial-(dis)similarity matrix · *N*-tuple simple stochastic · double stochastic and mutual probability matrices · Java language · chemical development kit · speed up · scalability

How to cite this article: C. R. García-Jacas, Y. Marrero-Ponce, L. Acevedo-Martínez, S. J. Barigye, J. R. Valdés-Martiní, E. Contreras-Torres, *J. Comput. Chem.* **2014**, *35*, 1395–1409. DOI: 10.1002/jcc.23640

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

[1] R. Todeschini, V. Consonni, Molecular Descriptors for Chemoinformatics; Wiley-VCH: Weinheim, **2009**.

[2] L. Hall, G. Kellogg, D. Haney, In http://www.molconn.com/; Hall Associates Consulting: Quincy, MA, **1991**.

[3] G. Cruciani, M. Pastor, W. Guba, *Eur. J. Pharm. Sci.* **2000**, *11*(Suppl 2), S29.

[4] A. Mauri, V. Consonni, M. Pavan, R. Todeschini, *Match* **2006**, *56*, 237.

[5] Z. R. Li, L. Y. Han, Y. Xue, C. W. Yap, H. Li, L. Jiang, Y. Z. Chen, *Biotechnol. Bioeng.* **2007**, *97*, 389.

[6] H. Hong, Q. Xie, W. Ge, F. Qian, H. Fang, L. Shi, Z. Su, R. Perkins,W. Tong, *J. Chem. Inf. Comput. Sci.* **2008**, *48*, 1337.

[7] H. Georg, In http://www.ra.cs.uni-tuebingen.de/software/bluedesc/welcome_ehtml; University of Tübingen: Tübingen, Germany, **2008**.

[8] R. Guha, Indiana. In http://www.rguha.net/code/java/cdkdeschtml.

[9] C. W. Yap, *J. Comput. Chem.* **2011**, *32*, 1466.

[10] Y. Marrero-Ponce, C. R. García-Jacas, S. J. Barigye, J. R. Valdés-Martiní, O. M. Rivera-Borroto, R. W. Pino-Urias, N. Cubillán, Y. Alvarado, *J. Curr. Bioinform*. (in press).

[11] C. R. García-Jacas, Y. Marrero-Ponce, S. J. Barigye, J. R. Valdés-Martiní, O. M. Rivera-Borroto, J. O. Verbel, *Curr. Drug Metabol. (in press).*

[12] Y. Marrero-Ponce, *Bioorg. Med. Chem.* **2004**, *12*, 6351.

[13] Y. Marrero-Ponce, F. Torrens,Y. J. Alvarado, R. Rotondo, *J. Comput. Aided Mol. Des.* **2006**, *20*, 685.

[14] Y. Marrero Ponce, *Molecules* **2003**, *8*, 687.

[15] J. A. Castillo-Garit, Y. Marrero-Ponce, F. Torrens, *Bioorg. Med. Chem.* **2006**, *14*, 2398.

[16] Y. Marrero-Ponce, F. Torrens, R. García-Domenech, S. E. Ortega-Broche, V. Romero Zaldivar, *J. Math. Chem.* **2008**, *44*, 650.

[17] Y. Marrero-Ponce, E. Martínez-Albelo, G. Casañola-Martín, J. A. Castillo-Garit, Y. Echevería-Díaz, V. Zaldivar, J. Tygat, J. Borges, R. García-Domenech, F. Torrens, F. Pérez-Giménez, *Mol. Divers.* **2010**, *14*, 731.

[18] Y. Marrero-Ponce, A. Huesca-Guillén, F. Ibarra-Velarde, *J. Mol. Struct. (THEOCHEM)* **2005**, *717*, 67.

[19] Y. Marrero-Ponce, J. A. Castillo-Garit, E. Olazabal, H. S. Serrano, A. Morales, N. Castañedo, F. Ibarra-Velarde, A. Huesca-Guillen, A. M. Sánchez, F. Torrens, E. A. Castro, *Bioorg. Med. Chem*. **2005**, *13*, 1005.

[20] J. A. Castillo-Garit, Y. Marrero-Ponce, F. Torrens,R. Rotondo, *J. Mol. Graph. Model.* **2007**, *26*, 32.

[21] J. W. Godden, F. L. Stahura, J. Bajorath, *J. Chem. Inf. Comput. Sci.* **2000**, *40*, 796.

[22] J. W. Godden, J. Bajorath, *J. Chem. Inf. Comput. Sci.* **2002**, *42*, 87.

[23] K. V. Mardia, J. T. Kent, J. M. Bibby, Multivariate Analysis; Academic Press: London, **1979**.

[24] R. D. Cramer, D. E. Patterson, J. D. Bunce, *J. Am. Chem. Soc.* **1988**, *110*, 5959.

[25] J. J. Sutherland, L. A. O'Brien, D. F. Weaver, *J. Med. Chem.* **2004**, *47*, 5541.

[26] R. W. Johnson, C. H. Huang, J. R. Johnson, *J. Supercomput.* **1991**, *5*, 189.

[27] D. Hestenes, G. Sobczyk, *Linear and Multilinear Functions*; Springer Netherlands, 1984.

[28] R. Todeschini, V. Consonni, *MATCH Commun. Math. Comput. Chem.* **2010**, *64*, 359.

[29] A. T. Balaban, *J. Chem. Inf. Comput. Sci.* **1994**, *34*, 398.

[30] J. A. Castillo-Garit, O. Martinez-Santiago,Y. Marrero-Ponce, G. M. Casañola-Martín, F. Torrens, *Chem. Phys. Lett.* **2008**, *464*, 107.

[31] A. K. Ghose, V. N. Viswanadhan, J. J. Wendoloski, *J. Phys. Chem. A* **1998**, *102*, 3762.

[32] A. T. Balaban, Ed., From Chemical Topology to Three-Dimensional Geometry; Plenum Press: New York, **1997**.

[33] A. T. Balaban, Steric Fit in Quantitative Structure-Activity Relations; Springer-Verlag: Berlin and New York, **1980**.

[34] J. Gasteiger, M. Marsili, *Tetrahedron* **1980**, *36*, 3219.

[35] P. Ertl, B. Rohde, P. Selzer, *J. Med. Chem.* **2000**, *43*, 3714.

[36] T. K. Ghanty, S. K. Ghosh, *J. Phys. Chem.* **1993**, *97*, 4951.

[37] J. Cioslowski, M. Martinov, *J. Chem. Phys.* **1994**, *101*, 366.

[38] S. J. Barigye, Y. Marrero-Ponce, Y. Martínez López, L. M. Artiles Martínez, R. W. Pino-Urias, O. Martínez Santiago, F. Torrens, *J. Comput. Chem.* **2013**, *34*, 259.

[39] S. J. Barigye, Y. Marrero-Ponce, O. M. Santiago, Y. M. López, F. Torrens, *Curr. Comput. Aided Drug. Des.* **2013**, *9*, 164.

[40] Y. Marrero-Ponce, O. Santiago, Y. López, S. Barigye, F. Torrens, *J. Comput. Aided Mol. Des.* **2012**, *26*, 1229.

[41] R. Todeschini, V. Consonni, Molecular Descriptors for Chemoinformatics, Vol. 1: Alphabetical Listing/Vol. 2: Appendices, References; Wiley-VCH: Weinheim, **2009**.

[42] A. C. Good, S. S. So, W. G. Richards, *J. Med. Chem.* **1993**, *36*, 433.

[43] G. Gasteiger, J. Sadowski, J. Schuur, P. Selzer, L. Steinhauer, V. Steinhauer, *J. Chem. Inf. Comput. Sci.* **1996**, *36*, 1030.

[44] B. Bogdanov, S. Nikolic, N. Trinajstic, *J. Math. Chem.* **1989**, *3*, 299.

[45] H. González-Díaz, E. Uriarte, *Bioorg. Med. Chem. Lett.* **2005**, *15*, 5088.

[46] R. R. de Armas, H. G. Díaz, R. Molina, E. Uriarte, *Proteins: Struct. Funct. Bioinform.* **2004**, *56*, 715.

[47] R. Carbo-Dorca, *Int. J. Quantum Chem.* **2000**, *79*, 163.

[48] R. Sinkhorn, *Ann. Math. Stat.* **1964**, *35*, 876.

[49] R. Sinkhorn, P. Knopp, *Pac. J. Math.* **1967**, *21*, 343.

[50] C. Steinbeck, Y. Q. Han, S. Kuhn, O. Horlacher, E. Luttmann, E. L. Willighagen, *J. Chem. Inf. Comput. Sci.* **2003**, *43*, 493.

[51] R. Todeschini, V. Consonni, A. Mauri, M. Pavan, R. Leardi, In Data Handling in Science and Technology; Leardi, R., Ed.; Elsevier: Amsterdan, The Netherlands, **2003**; pp. 141–167.

[52] G. Holmes, A. Donkin, I. H. Witten, Intelligent Information Systems, 1994. Proceedings of the 1994 Second Australian and New Zealand Conference, Brisbane, Qld., IEEE, November 29–December 2, **1994**; pp. 357–361.

[53] D. Fourches, E. Muratov,A. Tropsha, *J. Chem. Inf. Comput. Sci.* **2010**, *50*, 1189.

[54] M. Olah, M. Mracec, L. Ostopovici, R. Rad, A. Bora, N. Hadaruga, I. Olah, M. Banda, Z. Simon, M. Mracec, *Chemoinformatics in Drug Discovery*, Wiley-VCH: New York, Vol. *1*; **2004**.

[55] D. Lea, Proceedings of the ACM 2000 Conference on Java Grande, San Francisco, California, USA, ACM, **2000**; pp. 36–43.

[56] R. D. Blumofe, C. E. Leiserson, *J. ACM* **1999**, *46*, 720.

Received: 18 February 2014
Revised: 22 April 2014
Accepted: 23 April 2014
Published online on 30 May 2014