



Journal Articles

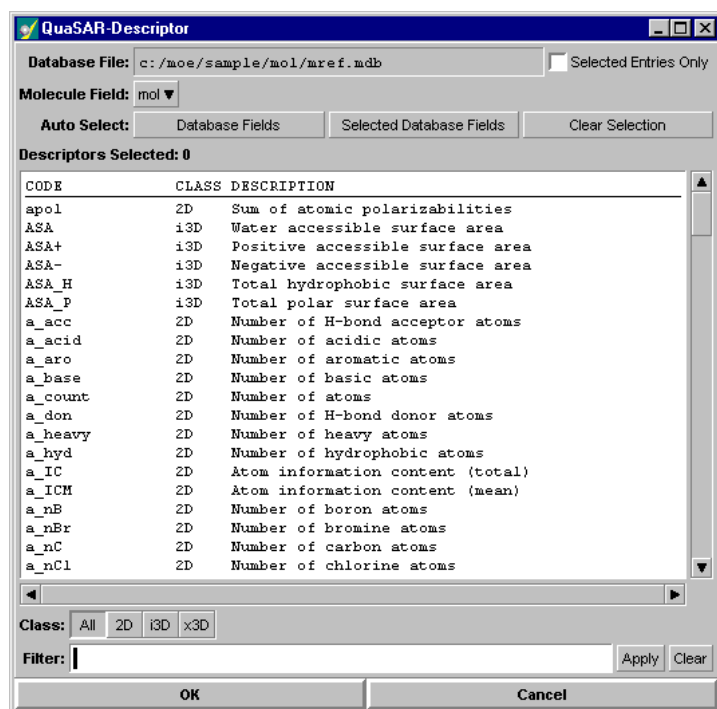
QuaSAR-Descriptor

- [Introduction](#)
- [2D Molecular Descriptors](#)
 - [Physical Properties](#)
 - [Subdivided Surface Areas](#)
 - [Atom Counts and Bond Counts](#)
 - [Kier&Hall Connectivity and Kappa Shape Indices](#)
 - [Adjacency and Distance Matrix Descriptors](#)
 - [Pharmacophore Feature Descriptors](#)
 - [Partial Charge Descriptors](#)
- [3D Molecular Descriptors](#)
 - [Potential Energy Descriptors](#)
 - [Surface Area, Volume and Shape Descriptors](#)
 - [Conformation Dependent Charge Descriptors](#)
- [Adding New Descriptors with SVL](#)
- [References](#)

Introduction

The purpose of QuaSAR-Descriptor is to calculate properties of molecules that serve as numerical descriptions or characterizations of molecules in other calculations such as QSAR, diversity analysis or combinatorial library design. In principle, because any molecular property may be used as a molecular descriptor, there is no single calculation procedure for QuaSAR-Descriptor. Rather, QuaSAR-Descriptor is a forum for the calculation of many descriptors.

A QuaSAR-Descriptor calculation proceeds as follows. Given a molecular database with a molecule field, a set of numerical properties will be calculated for each molecule and stored in the database. Every descriptor is given a unique name, or code, which identifies the descriptor. These codes are used as database field names. QuaSAR-Descriptor will overwrite fields with names identical to descriptor codes. When QuaSAR-Descriptor is invoked, the following panel appears:



This panel allows for selecting the list of descriptors to calculate. A keyword search facility can be used to restrict the list to particular descriptor families.

Descriptors are partitioned into *classes*. Each class indicates what is assumed by the descriptor calculators about the molecule presented:

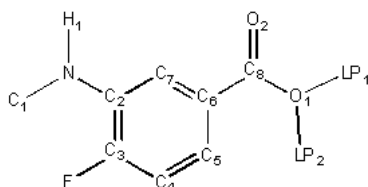
- **2D**. 2D descriptors only use the atoms and connection information of the molecule for the calculation. 3D coordinates and individual conformations are not considered.
- **i3D**. Internal 3D descriptors use 3D coordinate information about each molecule; however, they are invariant to rotations and translations of the conformation.
- **x3D**. External 3D descriptors also use 3D coordinate information but also require an absolute frame of reference (e.g., molecules docked into the same receptor).

2D Molecular Descriptors

2D molecular descriptors are defined to be numerical properties that can be calculated from the connection table representation of a molecule (*e.g.*, elements, formal charges and bonds, but not atomic coordinates). 2D descriptors are, therefore, not dependent on the conformation of a molecule and are most suitable for large database studies.

Notation and Terminology

Many descriptors make use of several fundamental quantities that can be computed from a chemical structure. This section will define these fundamental quantities. For purposes of illustration, the following chemical structure will be used:



The fundamental quantities of a chemical structure depend solely on the structure *as drawn*, i.e., no modifications to the structure are implied with the exception of the addition or subtraction of hydrogen atoms to full valence.

Z denotes the *atomic number* of an atom; lone pair pseudo-atoms (LP) are given an atomic number of 0. *Heavy atoms* are atoms that have an atomic number strictly greater than 1 (not H nor LP). A *trivial atom* is an LP pseudo-atom or a hydrogen with exactly one heavy neighbor. In the reference structure, H_1 , LP_1 and LP_2 are trivial.

The *hydrogen count*, h , of an atom is the number of hydrogens to which it is (or should be) attached. This count includes all hydrogen atoms that are necessary to fill valence. In the reference structure, F has $h = 0$, N has $h = 1$ and O_1 has $h = 1$.

The *heavy degree*, d , of an atom is the number of heavy atoms to which it is bonded. That is, d is the number of bonded neighbors of the atom in the hydrogen suppressed graph. In the reference structure, F has $d = 1$, C_6 has $d = 3$ and N has $d = 2$.

Physical Properties

The following physical properties can be calculated from the connection table (with no dependence on conformation) of a molecule:

Code	Description
apol	Sum of the atomic polarizabilities (including implicit hydrogens) with polarizabilities taken from [CRC 1994].
bpol	Sum of the absolute value of the difference between atomic polarizabilities of all bonded atoms in the molecule (including implicit hydrogens) with polarizabilities taken from [CRC 1994].
FCharge	Total charge of the molecule (sum of formal charges).
mr	Molecular refractivity (including implicit hydrogens). This property is calculated from an 11 descriptor linear model [MREF 1998] with $r^2 = 0.997$, RMSE = 0.168 on 1,947 small molecules.
SMR	Molecular refractivity (including implicit hydrogens). This property is an atomic contribution model [Crippen 1999] that assumes the correct protonation state (washed structures). The model was trained on ~7000 structures and results may vary from the mr descriptor.
Weight	Molecular weight (including implicit hydrogens) with atomic weights taken from [CRC 1994].
logP(o/w)	Log of the octanol/water partition coefficient (including implicit hydrogens). This property is calculated from a linear atom type model [LOGP 1998] with $r^2 = 0.931$, RMSE=0.393 on 1,847 molecules.
SlogP	Log of the octanol/water partition coefficient (including implicit hydrogens). This property is an atomic contribution model [Crippen 1999] that calculates logP from the given structure; i.e., the correct protonation state (washed structures). Results may vary from the logP(o/w) descriptor. The training set for SlogP was ~7000 structures.
vdw_vol	van der Waals volume calculated using a connection table approximation.
density	Molecular mass density: weight divided by vdw_vol.
vdw_area	Area of van der Waals surface calculated using a connection table approximation.

Subdivided Surface Areas

The Subdivided Surface Areas are descriptors based on an approximate accessible van der Waals surface area calculation for each atom, v_i along with some other atomic property, p_i . The v_i is calculated using a connection table approximation. Each descriptor in a series is defined to be the sum of the v_i over all atoms, i such that p_i is in a specified range $(a,b]$.

In the descriptions to follow, L_i denotes the contribution to $\log P(o/w)$ for atom i as calculated in the $slogP$ descriptor [Crippen 1999]. R_i denotes the contribution to Molar Refractivity for atom i as calculated in the SMR descriptor [Crippen 1999]. The ranges were determined by percentile subdivision over a large collection of compounds.

Code	Description
$slogP_VSA0$	Sum of v_i such that $L_i \leq -0.4$.
$slogP_VSA1$	Sum of v_i such that L_i is in $(-0.4, -0.2]$.
$slogP_VSA2$	Sum of v_i such that L_i is in $(-0.2, 0]$.
$slogP_VSA3$	Sum of v_i such that L_i is in $(0, 0.1]$.
$slogP_VSA4$	Sum of v_i such that L_i is in $(0.1, 0.15]$.
$slogP_VSA5$	Sum of v_i such that L_i is in $(0.15, 0.20]$.
$slogP_VSA6$	Sum of v_i such that L_i is in $(0.20, 0.25]$.
$slogP_VSA7$	Sum of v_i such that L_i is in $(0.25, 0.30]$.
$slogP_VSA8$	Sum of v_i such that L_i is in $(0.30, 0.40]$.
$slogP_VSA9$	Sum of v_i such that $L_i > 0.40$.
SMR_VSA0	Sum of v_i such that R_i is in $[0, 0.11]$.
SMR_VSA1	Sum of v_i such that R_i is in $(0.11, 0.26]$.
SMR_VSA2	Sum of v_i such that R_i is in $(0.26, 0.35]$.
SMR_VSA3	Sum of v_i such that R_i is in $(0.35, 0.39]$.
SMR_VSA4	Sum of v_i such that R_i is in $(0.39, 0.44]$.
SMR_VSA5	Sum of v_i such that R_i is in $(0.44, 0.485]$.
SMR_VSA6	Sum of v_i such that R_i is in $(0.485, 0.56]$.
SMR_VSA7	Sum of v_i such that $R_i > 0.56$.

Atom Counts and Bond Counts

The atom count and bond count descriptors are functions of the counts of atoms and bonds (subdivided according to various criteria).

Code	Description
a_aro	Number of aromatic atoms.
a_count	Number of atoms (including implicit hydrogens). This is calculated as the sum of $(1 + h_i)$ over all non-trivial atoms i .
a_heavy	Number of heavy atoms $\#\{Z_i \mid Z_i > 1\}$
a_ICM	

	Atom information content (mean). This is the entropy of the element distribution in the molecule (including implicit hydrogens but not lone pair pseudo-atoms). Let n_i be the number of occurrences of atomic number i in the molecule. Let $p_i = n_i / n$ where n is the sum of the n_i . The value of a_ICM is the negative of the sum over all i of $p_i \log p_i$.
a_IC	Atom information content (total). This is a_ICM times n (as defined in the definition of a_ICM).
a_nH	Number of hydrogen atoms (including implicit hydrogens). This is calculated as the sum of h_i over all non-trivial atoms i plus the number of non-trivial hydrogen atoms.
a_nB	Number of boron atoms: $\#\{Z_i \mid Z_i = 5\}$
a_nC	Number of carbon atoms: $\#\{Z_i \mid Z_i = 6\}$
a_nN	Number of nitrogen atoms: $\#\{Z_i \mid Z_i = 7\}$
a_nO	Number of oxygen atoms: $\#\{Z_i \mid Z_i = 8\}$
a_nF	Number of fluorine atoms: $\#\{Z_i \mid Z_i = 9\}$
a_nP	Number of phosphorus atoms: $\#\{Z_i \mid Z_i = 15\}$
a_nS	Number of sulfur atoms: $\#\{Z_i \mid Z_i = 16\}$
a_nCl	Number of chlorine atoms: $\#\{Z_i \mid Z_i = 17\}$
a_nBr	Number of bromine atoms: $\#\{Z_i \mid Z_i = 35\}$
a_nI	Number of iodine atoms: $\#\{Z_i \mid Z_i = 53\}$
b_1rotN	Number of rotatable single bonds. A bond is rotatable if it is not in a ring, and neither atom of the bond is such that $(d_i + h_i) < 2$.
b_1rotR	Fraction of rotatable single bonds: b_1rotN divided by b_count .
b_ar	Number of aromatic bonds.
b_count	Number of bonds (including implicit hydrogens). This is calculated as the sum of $(d_i/2 + h_i)$ over all non-trivial atoms i .
b_double	Number of double bonds. Aromatic bonds are not considered to be double bonds.
b_heavy	Number of bonds between heavy atoms
b_rotN	Number of rotatable bonds. A bond is rotatable if it is not in a ring, and neither atom of the bond is such that $(d_i + h_i) < 2$.
b_rotR	Fraction of rotatable bonds: b_rotN divided by b_count .
b_single	Number of single bonds (including implicit hydrogens). Aromatic bonds are not considered to be single bonds.
b_triple	Number of triple bonds. Aromatic bonds are not considered to be triple bonds.
vAdjMa	Vertex adjacency information (magnitude): $1 + \log_2 m$ where m is the number of heavy-heavy bonds. If m is zero, then zero is returned.
vAdjEq	Vertex adjacency information (equality): $-(1-f)\log_2(1-f) - f\log_2 f$ where $f = (n^2 - m) / n^2$, n is the number of heavy atoms and m is the number of heavy-heavy bonds. If f is not in the open interval $(0,1)$, then 0 is returned.

Kier&Hall Connectivity and Kappa Shape Indices

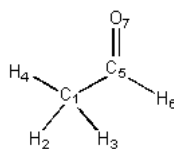
For a heavy atom i let $v_i = (p_i - h_i) / (Z_i - p_i - 1)$ where p_i is the number of s and p valence electrons of atom i . The Kier and Hall chi connectivity indices are calculated from the d_i and v_i values. The Kier and Hall kappa molecular shape indices [Hall 1991] compare the molecular graph with minimal and maximal molecular graphs, and are intended to capture different aspects of molecular shape. In the following description, n denotes the number of atoms in the hydrogen suppressed graph, m is the number of bonds in the hydrogen suppressed graph and a is the sum of $(r_i/r_c - 1)$ where r_i is the covalent radius of atom i and r_c is the covalent radius of a carbon atom. Also, let p_2 denote the number of paths of length 2 and let p_3 denote the number of paths of length 3.

Code	Description
chi0	Atomic connectivity index (order 0) from [Hall 1991] and [Hall 1997]. This is calculated as the sum of $1/\sqrt{d_i}$ over all heavy atoms i with $d_i > 0$.
chi0_C	Carbon connectivity index (order 0). This is calculated as the sum of $1/\sqrt{d_i}$ over all carbon atoms i with $d_i > 0$.
chi1	Atomic connectivity index (order 1) from [Hall 1991] and [Hall 1997]. This is calculated as the sum of $1/\sqrt{d_i d_j}$ over all bonds between heavy atoms i and j where $i < j$.
chi1_C	Carbon connectivity index (order 1). This is calculated as the sum of $1/\sqrt{d_i d_j}$ over all bonds between carbon atoms i and j where $i < j$.
chi0v	Atomic valence connectivity index (order 0) from [Hall 1991] and [Hall 1997]. This is calculated as the sum of $1/\sqrt{v_i}$ over all heavy atoms i with $v_i > 0$.
chi0v_C	Carbon valence connectivity index (order 0). This is calculated as the sum of $1/\sqrt{v_i}$ over all carbon atoms i with $v_i > 0$.
chi1v	Atomic valence connectivity index (order 1) from [Hall 1991] and [Hall 1997]. This is calculated as the sum of $1/\sqrt{v_i v_j}$ over all bonds between heavy atoms i and j where $i < j$.
chi1v_C	Carbon valence connectivity index (order 1). This is calculated as the sum of $1/\sqrt{v_i v_j}$ over all bonds between carbon atoms i and j where $i < j$.
Kier1	First kappa shape index: $(n-1)^2 / m^2$ [Hall 1991]
Kier2	Second kappa shape index: $(n-1)^2 / m^2$ [Hall 1991]
Kier3	Third kappa shape index: $(n-1)(n-3)^2 / p_3^2$ for odd n , and $(n-3)(n-2)^2 / p_3^2$ for even n [Hall 1991]
KierA1	First alpha modified shape index: $s(s-1)^2 / m^2$ where $s = n + a$ [Hall 1991]
KierA2	Second alpha modified shape index: $s(s-1)^2 / m^2$ where $s = n + a$ [Hall 1991]
KierA3	Third alpha modified shape index: $(n-1)(n-3)^2 / p_3^2$ for odd n , and $(n-3)(n-2)^2 / p_3^2$ for even n where $s = n + a$ [Hall 1991]
KierFlex	Kier molecular flexibility index: $(\text{KierA1})(\text{KierA2}) / n$ [Hall 1991]
zagreb	Zagreb index: the sum of d_i^2 over all heavy atoms i .

Adjacency and Distance Matrix Descriptors

The *adjacency matrix*, M , of a chemical structure is defined by the elements $[M_{ij}]$ where M_{ij} is 1 if atoms i and j are bonded and zero otherwise. The *distance matrix*, D , of a chemical structure is defined by the elements $[D_{ij}]$ where D_{ij} is the length of the shortest path from atoms i to j ; zero is used if atoms i and j are not part of the same connected component. The adjacency matrix of CH₃CH=O is displayed on the left and its distance matrix is displayed on the right (below):

C1	0	1	1	1	0	0	0	1	1	1	1	2	2
H2	1	0	0	0	0	0	0	1	0	2	2	2	3
H3	1	0	0	0	0	0	0	1	2	0	2	2	3
H4	1	0	0	0	0	0	0	1	2	2	0	2	3
C5	1	0	0	0	0	1	1	1	2	2	2	0	1
H6	0	0	0	0	1	0	0	2	3	3	3	1	0
O7	0	0	0	0	1	0	0	2	3	3	3	1	2



The following descriptors are calculated from the distance and adjacency matrices of the heavy atoms:

Code	Description
balabanJ	Balaban's connectivity topological index [Balaban 1982].
diameter	Largest value in the distance matrix [Petitjean 1992].
petitjean	Value of $(\text{diameter} - \text{radius}) / \text{diameter}$ as defined in [Petitjean 1992].
radius	If r_i is the largest matrix entry in row i of the distance matrix D , then the radius is defined as the smallest of the r_i [Petitjean 1992].
vDistEq	If m is the sum of the distance matrix entries then vDistEq is defined to be the sum of $\log_2 m - p_i \log_2 p_i / m$ where p_i is the number of distance matrix entries equal to i .
vDistMa	If m is the sum of the distance matrix entries then vDistMa is defined to be the sum of $\log_2 m - D_{ij} \log_2 D_{ij} / m$ over all i and j .
weinerPath	Wiener path number: half the sum of all the distance matrix entries as defined in [Balaban 1979] and [Wiener 1947].
weinerPol	Wiener polarity number: half the sum of all the distance matrix entries with a value of 3 as defined in [Balaban 1979].

Pharmacophore Feature Descriptors

The Pharmacophore Atom Type descriptors consider only the heavy atoms of a molecule and assign a type to each atom (using a rule-based system). That is, hydrogens are suppressed during the calculation. The feature set is Donor, Acceptor, Polar (both Donor and Acceptor), Positive (base), Negative (acid), Hydrophobe and Other. Assignments may take into account implied protonation, deprotonation, keto/enol considerations and tautomerism at a biologically relevant pH. For example, -COOH will be typed in its deprotonated form regardless of how the structure is stored.

Code	Description
a_acc	Number of hydrogen bond acceptor atoms (not counting acidic atoms but counting atoms that are both hydrogen bond donors and acceptors such as -OH).
a_acid	Number of acidic atoms.
a_base	Number of basic atoms.
a_don	Number of hydrogen bond donor atoms (not counting basic atoms but counting atoms that are both hydrogen bond donors and acceptors such as -

	OH).
a_hyd	Number of hydrophobic atoms.
vsa_acc	Approximation to the sum of VDW surface areas of pure hydrogen bond acceptors (not counting acidic atoms and atoms that are both hydrogen bond donors and acceptors such as -OH).
vsa_acid	Approximation to the sum of VDW surface areas of acidic atoms.
vsa_base	Approximation to the sum of VDW surface areas of basic atoms.
vsa_don	Approximation to the sum of VDW surface areas of pure hydrogen bond donors (not counting basic atoms and atoms that are both hydrogen bond donors and acceptors such as -OH).
vsa_hyd	Approximation to the sum of VDW surface areas of hydrophobic atoms.
vsa_other	Approximation to the sum of VDW surface areas of atoms typed as "other".
vsa_pol	Approximation to the sum of VDW surface areas of polar (both hydrogen bond donors and acceptors) atoms (such as -OH).

Partial Charge Descriptors

Descriptors that depend on the partial charge of each atom of a chemical structure require calculation of those partial charges. An unfortunate complication is the fact that there are numerous methods of calculating partial charges. Rather than enforce a particular method, MOE provides several versions of most of the charge-dependent descriptors. The only difference between these variants is the source of the partial charges. The following variants are supported: PEOE, Q (described below).

PEOE. The Partial Equalization of Orbital Electronegativities (PEOE) method of calculating atomic partial charges [Gasteiger 1980] is a method in which charge is transferred between bonded atoms until equilibrium. To guarantee convergence, the amount of charge transferred at each iteration is damped with an exponentially decreasing scale factor. The amount of charge transferred, dq_{ij} , between atoms i and j when $X_i > X_j$ is

$$dq_{ij} = (1/2^k) (X_i - X_j) / X_j^+$$

where X_j^+ is the electronegativity of the positive ion of atom j ; X_i is the electronegativity of atom i (quadratically dependent on partial charge); and k is the iteration number of the algorithm. The PEOE charges depend only on the connectivity of the input structures: elements, formal charges and bond orders. Descriptors using the PEOE charges are prefixed with PEOE_.

Q. Descriptors prefixed with q_ use the partial charges stored with each structure in the database. In other words, no partial charge calculation is made and it is assumed that some external program has been used to calculate the atomic partial charges. This dependence can be a subtle source of error if, for example, the wrong charges are stored when descriptors are recalculated (e.g., when evaluating QSAR models on novel structures).

Let q_i denote the partial charge of atom i as defined above. Let v_i be the van der Waals surface area of atom i (as calculated by a connection table approximation). The following descriptors are calculated:

Code	Description
Q_PC+ PEOE_PC+	Total positive partial charge: the sum of the positive q_i . Q_PC+ is identical to PC+ which has been retained for compatibility.
Q_PC- PEOE_PC-	Total negative partial charge: the sum of the negative q_i . Q_PC- is identical to PC- which has been retained for compatibility.

Q_RPC+ PEOE_RPC+	Relative positive partial charge: the largest positive q_i divided by the sum of the positive q_i . Q_RPC+ is identical to RPC+ which has been retained for compatibility.
Q_PRC- PEOE_RPC-	Relative negative partial charge: the smallest negative q_i divided by the sum of the negative q_i . Q_RPC- is identical to RPC- which has been retained for compatibility.
Q_VSA_POS PEOE_VSA_POS	Total positive van der Waals surface area. This is the sum of the v_i such that q_i is non-negative. The v_i are calculated using a connection table approximation.
Q_VSA_NEG PEOE_VSA_NEG	Total negative van der Waals surface area. This is the sum of the v_i such that q_i is negative. The v_i are calculated using a connection table approximation.
Q_VSA_PPOS PEOE_VSA_PPOS	Total positive polar van der Waals surface area. This is the sum of the v_i such that q_i is greater than 0.2. The v_i are calculated using a connection table approximation.
Q_VSA_PNEG PEOE_VSA_PNEG	Total negative polar van der Waals surface area. This is the sum of the v_i such that q_i is less than -0.2. The v_i are calculated using a connection table approximation.
Q_VSA_HYD PEOE_VSA_HYD	Total hydrophobic van der Waals surface area. This is the sum of the v_i such that $ q_i $ is less than or equal to 0.2. The v_i are calculated using a connection table approximation.
Q_VSA_POL PEOE_VSA_POL	Total polar van der Waals surface area. This is the sum of the v_i such that $ q_i $ is greater than 0.2. The v_i are calculated using a connection table approximation.
Q_VSA_FPOS PEOE_VSA_FPOS	Fractional positive van der Waals surface area. This is the sum of the v_i such that q_i is non-negative divided by the total surface area. The v_i are calculated using a connection table approximation.
Q_VSA_FNEG PEOE_VSA_FNEG	Fractional negative van der Waals surface area. This is the sum of the v_i such that q_i is negative divided by the total surface area. The v_i are calculated using a connection table approximation.
Q_VSA_FPPOS PEOE_VSA_FPPOS	Fractional positive polar van der Waals surface area. This is the sum of the v_i such that q_i is greater than 0.2 divided by the total surface area. The v_i are calculated using a connection table approximation.
Q_VSA_FPNEG PEOE_VSA_FPNEG	Fractional negative polar van der Waals surface area. This is the sum of the v_i such that q_i is less than -0.2 divided by the total surface area. The v_i are calculated using a connection table approximation.
Q_VSA_FHYD PEOE_VSA_FHYD	Fractional hydrophobic van der Waals surface area. This is the sum of the v_i such that $ q_i $ is less than or equal to 0.2 divided by the total surface area. The v_i are calculated using a connection table approximation.
Q_VSA_FPOL PEOE_VSA_FPOL	Fractional polar van der Waals surface area. This is the sum of the v_i such that $ q_i $ is greater than 0.2 divided by the total surface area. The v_i are calculated using a connection table approximation.
PEOE_VSA+6	Sum of v_i where q_i is greater than 0.3.
PEOE_VSA+5	Sum of v_i where q_i is in the range [0.25,0.30).
PEOE_VSA+4	

	Sum of v_i where q_i is in the range [0.20,0.25).
PEOE_VSA+3	Sum of v_i where q_i is in the range [0.15,0.20).
PEOE_VSA+2	Sum of v_i where q_i is in the range [0.10,0.15).
PEOE_VSA+1	Sum of v_i where q_i is in the range [0.05,0.10).
PEOE_VSA+0	Sum of v_i where q_i is in the range [0.00,0.05).
PEOE_VSA-0	Sum of v_i where q_i is in the range [-0.05,0.00).
PEOE_VSA-1	Sum of v_i where q_i is in the range [-0.10,-0.05).
PEOE_VSA-2	Sum of v_i where q_i is in the range [-0.15,-0.10).
PEOE_VSA-3	Sum of v_i where q_i is in the range [-0.20,-0.15).
PEOE_VSA-4	Sum of v_i where q_i is in the range [-0.25,-0.20).
PEOE_VSA-5	Sum of v_i where q_i is in the range [-0.30,-0.25).
PEOE_VSA-6	Sum of v_i where q_i is less than -0.30.

3D Molecular Descriptors

There are two types of 3D molecular descriptors: those that depend on internal coordinates only and those that depend on absolute orientation. 3D molecular descriptors are classified as "i3D" for internal coordinate dependent 3D and "x3D" for external coordinate dependent. A good example is the dipole moment: the magnitude of the dipole moment does not depend on absolute orientation in space; however, the x component of the dipole moment does depend on absolute orientation.

Potential Energy Descriptors

The energy descriptors use the MOE potential energy model to calculate energetic quantities from stored 3D conformations. Most of the energy descriptors belong to the i3D class; that is, they depend on internal coordinates alone and not on an external reference frame. Descriptors that rely on an external reference frame are clearly indicated in the list below.

Code	Description
E	Value of the potential energy. The state of all term enable flags will be honored (in addition to the term weights). This means that the current potential setup accurately reflects what will be calculated.
E_ang	Angle bend potential energy. In the Potential Setup panel, the term enable flag is ignored, but the term weight is applied.
E_ele	Electrostatic component of the potential energy. In the Potential Setup panel, the term enable flag is ignored, but the term weight is applied.
E_nb	Value of the potential energy with all non-bonded terms disabled. Thus, the state of the non-bonded term enable flags will be honored (in addition to the term weights).
E_oop	Out-of-plane potential energy. In the Potential Setup panel, the term enable flag is ignored, but the term weight is applied.
E_sol	Solvation energy. In the Potential Setup panel, the term enable flag is ignored, but the term weight is applied.
E_stb	Bond stretch-bend cross-term potential energy. In the Potential Setup panel, the term enable flag is ignored, but the term weight is applied.

E_str	Bond stretch potential energy. In the Potential Setup panel, the term enable flag is ignored, but the term weight is applied.
E_strain	Local strain energy: the current energy minus the value of the energy at a near local minimum. The current energy is calculated as for the ϵ descriptor. The local minimum energy is the value of the ϵ descriptor after first performing an energy minimization. Current chirality is preserved and charges are left undisturbed during minimization. The structure in the database is not modified (results of the minimization are discarded).
E_tor	Torsion (proper and improper) potential energy. In the Potential Setup panel, the term enable flag is ignored, but the term weight is applied.
E_vdw	van der Waals component of the potential energy. In the Potential Setup panel, the term enable flag is ignored, but the term weight is applied.
E_rele	Electrostatic interaction energy (external reference frame: x3d) between the stored molecule and the atoms currently loaded. In the Potential Setup panel, the term enable flag is ignored, but the term weight is applied. Partial charges are assumed to be correct in the database molecule as well as the currently loaded atoms.
E_rsol	Solvation free energy difference (external reference frame: x3d). Let L be the free energy of solvation of the stored molecule (ligand), R be the free energy of solvation of the atoms currently loaded (receptor), and G be the free energy of solvation of the RL complex. Consequently, the returned value is $G - L - R$. In the Potential Setup panel, the term enable flag is ignored, but the term weight is applied. Partial charges are assumed to be correct in the database molecule as well as the currently loaded atoms.
E_rvdw	van der Waals interaction energy (external reference frame: x3d) between the stored molecule and the atoms currently loaded. In the Potential Setup panel, the term enable flag is ignored, but the term weight is applied.

Surface Area, Volume and Shape Descriptors

The following descriptors depend on the structure connectivity and conformation:

Code	Description
ASA	Water accessible surface area calculated using a radius of 1.4 Å for the water molecule. A polyhedral representation is used for each atom in calculating the surface area.
dens	Mass density: molecular weight divided by van der Waals volume as calculated in the <code>vo1</code> descriptor.
glob	Globularity, or inverse condition number (smallest eigenvalue divided by the largest eigenvalue) of the covariance matrix of atomic coordinates. A value of 1 indicates a perfect sphere while a value of 0 indicates a two- or one-dimensional object.
pmi	Principal moment of inertia.
pmiX	x component of the principal moment of inertia (external coordinates).
pmiY	y component of the principal moment of inertia (external coordinates).
pmiZ	z component of the principal moment of inertia (external coordinates).
rgyr	Radius of gyration.
std_dim1	Standard dimension 1: the square root of the largest eigenvalue of the

	covariance matrix of the atomic coordinates. A standard dimension is equivalent to the standard deviation along a principal component axis.
std_dim2	Standard dimension 2: the square root of the second largest eigenvalue of the covariance matrix of the atomic coordinates. A standard dimension is equivalent to the standard deviation along a principal component axis.
std_dim3	Standard dimension 3: the square root of the third largest eigenvalue of the covariance matrix of the atomic coordinates. A standard dimension is equivalent to the standard deviation along a principal component axis.
vol	van der Waals volume calculated using a grid approximation (spacing 0.75 Å).
VSA	van der Waals surface area. A polyhedral representation is used for each atom in calculating the surface area.

Conformation Dependent Charge Descriptors

The following descriptors depend upon the stored partial charges of the molecules and their conformations. Accessible surface area refers to the water accessible surface area using a probe radius of 1.4 Å. Let q_i denote the partial charge of atom i .

Code	Description
ASA+	Water accessible surface area of all atoms with positive partial charge (strictly greater than 0).
ASA-	Water accessible surface area of all atoms with negative partial charge (strictly less than 0).
ASA_H	Water accessible surface area of all hydrophobic ($ q_i < 0.2$) atoms.
ASA_P	Water accessible surface area of all polar ($ q_i \geq 0.2$) atoms.
DASA	Absolute value of the difference between ASA+ and ASA-.
CASA+	Positive charge weighted surface area, ASA+ times $\max \{ q_i > 0 \}$ [Stanton 1990].
CASA-	Negative charge weighted surface area, ASA- times $\max \{ q_i < 0 \}$ [Stanton 1990].
DCASA	Absolute value of the difference between CASA+ and CASA- [Stanton 1990].
dipole	Dipole moment calculated from the partial charges of the molecule.
dipoleX	The x component of the dipole moment (external coordinates).
dipoleY	The y component of the dipole moment (external coordinates).
dipoleZ	The z component of the dipole moment (external coordinates).
FASA+	Fractional ASA+ calculated as $ASA+ / ASA$.
FASA-	Fractional ASA- calculated as $ASA- / ASA$.
FCASA+	Fractional CASA+ calculated as $CASA+ / ASA$.
FCASA-	Fractional CASA- calculated as $CASA- / ASA$.
FASA_H	Fractional ASA_H calculated as ASA_H / ASA .
FASA_P	Fractional ASA_P calculated as ASA_P / ASA .

Adding New Descriptors with SVL

Descriptor calculation is handled by a module that searches the MOE system for SVL functions satisfying a specific naming convention. Each such function is responsible for calculating a descriptor or family of related descriptors. Typically, such functions are located in their own SVL source code file which must be loaded in the system prior to running the QuaSAR applications. Adding a descriptor involves writing a file containing SVL functions for registering and calculating the descriptor value, and then loading that file into the system:

1. Create a file called, say, `mydesc.svl`, that conforms to the QuaSAR-Descriptor conventions described below.
2. Load the descriptor module into MOE with `load 'mydesc.svl'`. Note that if the QuaSAR-Descriptor panel is up when you load or re-load a descriptor file, the panel will not reflect the changes. Close the panel and re-open it to see the new descriptors. This also applies to other applications that use QuaSAR descriptors.

Here is an example of a descriptor file (explanations follow):

```
//
//      mydesc.svl          sample new descriptors
//

#set title  'My Descriptors'      // title of module
#set class  'QuaSAR'             // module class of descriptors

function QuaSAR_list_MyDescriptors [] = tr [
  [ 'Caro',    'Number of aromatic C',    '2D', [] ],
  [ 'C=O',     'Number of carbonyl C',    '2D', [] ]
];

function QuaSAR_calc_MyDescriptors [db_mol, codes, parms]
  local desc = zero codes;

  // load the database molecule into MOE as objects

  local [chains, molecule_name] = db_CreateMolecule db_mol;
  local atoms = cat cAtoms chains;

  // calculate the individual descriptors and assign
  // them to the corresponding positions in the return vector

  (desc | codes == 'C=O' ) = add sm_Match ['C=O', atoms];
  (desc | codes == 'Caro') = add sm_Match ['c',  atoms];

  oDestroy chains;          // destroy created objects
  return desc;
endfunction
```

The header of the module is typical of SVL program files: a comment header followed by SVL compiler directives:

```
//
//      mydesc.svl          sample new descriptors
//

#set title  'My Descriptors'      // title of module
#set class  'QuaSAR'             // module class of descriptors
```

The `#set title` directive assigns a title to the SVL module which will appear in the Modules and Task window and give some indication as to the contents of the source code file. The `#set class` directive assigns a class (group of related SVL files) to the module. Descriptor modules are usually put in the QuaSAR class. This ensures that all descriptor modules are listed together in the Modules and Tasks window.

The descriptor file must contain two global functions that, together, a) define the descriptor to the rest of the system; and b) calculate the descriptor when given a molecule. A naming convention is used to identify these functions (the SVL file can define other functions if needed):

- The function that declares the descriptors to the rest of the system must start with the prefix `QuaSAR_list_`. This function takes no parameters and returns a table of information detailing the set of descriptors that can be calculated with the associated calculation function.
- The function that calculates the descriptors must start with the prefix `QuaSAR_calc_`. This function is passed a molecule (in database format) and a set of descriptor codes to calculate. It must return the calculated descriptors.

The suffix of the list and calculate functions must be the same. Any set of characters can be used, but the two functions must be unique with respect to all other global symbols (*i.e.*, choose descriptive names). In the example file (`mydesc.svl` above), the list function is `QuaSAR_list_MyDescriptors`:

```
function QuaSAR_list_MyDescriptors [] = tr [
  [ 'Caro',    'Number of aromatic C',    '2D', [ ] ],
  [ 'C=O',     'Number of carbonyl C',    '2D', [ ] ]
];
```

List functions must return a table of data detailing which descriptors the calculate function can calculate. This table is a vector of lists of the form:

```
[code, description, class, parm]
```

Each of the elements of this vector must have the same length. The elements are interpreted as follows:

- `code(i)` is a token defining the descriptor code in the QuaSAR system. This identifier must be unique amongst all other descriptors. Example identifiers are 'chi0' and 'dipole'.
- `description(i)` is a token containing a short, one-line description of the descriptor which will appear in the QuaSAR-Descriptor panel. Examples are 'Principal moment of inertia' and 'Number of carbon atoms'.
- `class(i)` is a token containing the class of the descriptor and accepts values such as '2D', 'i3D' or 'x3D'.
- `parm(i)` is reserved for future use and should be set to the null vector [].

In the example, the list function is written so that each descriptor calculated by the calculation function is described on one line of the form:

```
[ 'Caro',    'Number of aromatic C',    '2D', [ ] ],
```

and the `tr` operator is used to convert this transposed representation into the correct form for the QuaSAR system.

The calculation function associated with the list function is `QuaSAR_list_MyDescriptors`. The association is created because of the common suffix `MyDescriptors`. A calculation function must be declared as:

```
function QuaSAR_calc_name [db_mol, codes, parms]
  // .... body of function ....
endfunction
```

where

`name`

Name used in the QuaSAR-List function. In this example, it would be `MyDescriptors`.

`db_mol`

SVL vector representation of a molecule in the form returned by `db_ExtractMolecule`.

`codes`

Specifies which descriptors are to be calculated and is in the form of a vector of tokens. The calculation function may assume that each of the passed codes is contained in the set advertised by the list function. Duplicate codes are allowed to appear in the passed `codes` vector.

parms

Reserved for future use and currently set to a vector of nulls equal in length to `codes`.

The calculation function must return a vector `desc` equal in length to `codes` such that `desc(i)` is the value of the descriptor specified by `code(i)`. The calculation function must be designed to accept more than one code at a time. In the example, the calculation function handles two descriptors. To handle multiple occurrences of descriptor codes, the following logic is typically used:

```
function QuaSAR_calc_MyDescriptors [db_mol, codes, parms]
  local desc = zero codes;

  // .... create molecule ....

  (desc | codes == 'C=O' ) = add sm_Match ['C=O', atoms];
  (desc | codes == 'Caro') = add sm_Match ['c', atoms];

  // .... destroy molecule ....

  return desc;
endfunction
```

The initialization of `desc` creates a zero vector of equal length to `codes`. Once the descriptors have been calculated, they are assigned to the correct locations with code of the form:

```
(desc | codes == 'mydesc') = value_of_mydesc;
```

The remainder of the calculation function handles the creation and destruction of the molecular objects in MOE. However, if the descriptor can be calculated solely from the `db_mol` argument, then there is no need to create molecular objects. The `a_heavy` descriptor (number of heavy atoms) is a good example of a descriptor that can be calculated directly from the `db_mol` parameter.

References

- [Balaban 1979] Balaban, A.T.; Five New Topological Indices for the Branching of Tree-Like Graphs; *Theoretica Chimica Acta*. 53, 355-375, (1979)
- [Balaban 1982] Balaban, A.T.; Highly Discriminating Distance-Based Topological Index; *Chemical Physics Letters*. Vol.89, No.5, 399-404, (1982).
- [CRC 1994] *CRC Handbook of Chemistry and Physics*. CRC Press (1994).
- [Crippen 1999] Wildman, S.A., Crippen, G.M.; Prediction of Physicochemical Parameters by Atomic Contributions. *J. Chem. Inf. Comput. Sci.* 39(5), 868-873, (1999).
- [Gasteiger 1980] Gasteiger, J., Marsali, M.; Iterative Partial Equalization of Orbital Electronegativity - A Rapid Access to Atomic Charges. *Tetrahedron*. 36, 3219, (1980).
- [Hall 1991] Hall, L.H., Kier, L.B.; The Molecular Connectivity Chi Indices and Kappa Shape Indices in Structure-Property Modeling. *Reviews of Computational Chemistry*. Vol 2, (1991).
- [Hall 1997] Hall, L.H., Kier, L.B.; The Nature of Structure-Activity Relationships and Their Relation to Molecular Connectivity. *Eur. J. Med. Chem. - Chimica Therapeutica*. 4, 307-312, (1997).
- [LOGP 1998] Labute, P.; MOE LogP(Octanol/Water) Model. *unpublished*. Source code in `$MOE/lib/avl/quasar.svl/q_logp.svl`, (1998).
- [MREF 1998] Labute, P.; MOE Molar Refractivity Model. *unpublished*. Source code in `$MOE/lib/avl/quasar.svl/q_mref.svl`, (1998)
- [Petitjean 1992] Petitjean, M.; Applications of the Radius-Diameter Diagram to the Classification of Topological and Geometrical Shapes of Chemical Compounds. *J. Chem. Inf. Comput. Sci.* 32, 331-337, (1992).

[Stanton 1990] Stanton D., Jurs P.; *Anal. Chem.* Vol.62, 2323 (1990).

[Wiener 1947] Wiener, H. Structural Determination of Paraffin Boiling Points. *Journal of the American Chemical Society*. Vol. 69, 17-20, (1947).

© 2016 Chemical Computing Group Inc. All rights reserved

www.chemcomp.com