

Calculate Descriptors

- [Introduction](#)
- [Running the Calculation](#)
- [2D Molecular Descriptors](#)
 - [Notation and Terminology](#)
 - [Physical Properties](#)
 - [Hueckel Theory Descriptors](#)
 - [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](#)
- [Protein Property Descriptors](#)
- [3D Molecular Descriptors](#)
 - [Potential Energy Descriptors](#)
 - [MOPAC Descriptors](#)
 - [Surface Area, Volume and Shape Descriptors](#)
 - [Conformation Dependent Charge Descriptors](#)
- [Adding New Descriptors with SVL](#)
- [SVL Commands](#)
- [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.

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).
- **Protein**. Physicochemical protein properties.

Warning! Most descriptors assume that the molecule stored in the database is drawn correctly, with the exception of potentially missing hydrogens. Counter-ions, salts, and other artifacts will be included in the descriptor calculations.

Warning! Both the i3D and the x3D descriptors **require that the database hold reasonable conformations of the stored molecules**. The MOE 3D converter can be used to create reasonable conformations and can be invoked with [Energy Minimize](#) in the Database Viewer. Alternatively, the conformations can be imported from some other 3D structure converter. Also, the 3D descriptors will not add or remove any atoms prior to calculation.

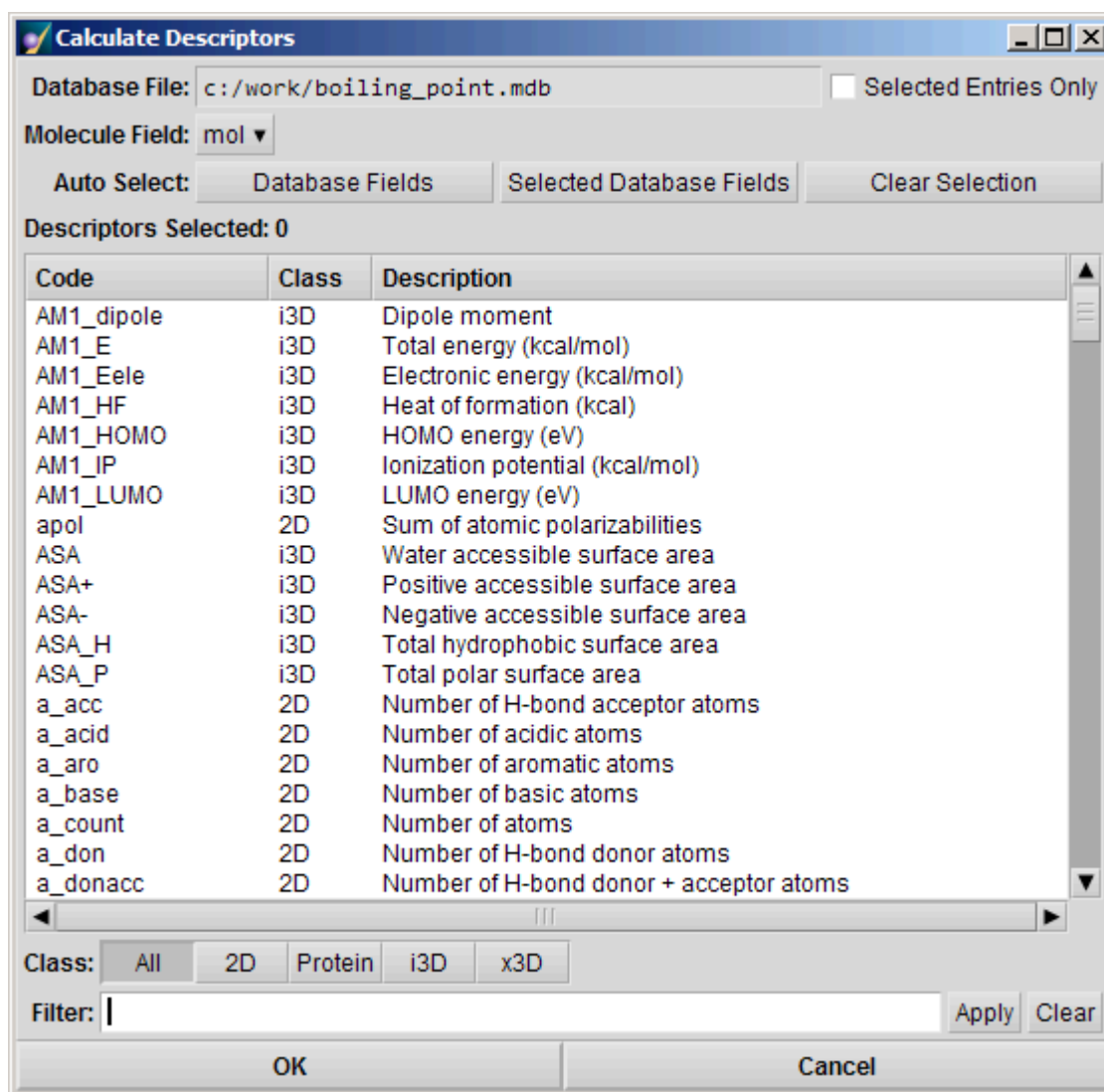
Running the Calculation

Unless interaction energy descriptors are to be calculated, it is a good idea to clear MOE of any molecular objects. Some descriptors require this condition while others are slowed down if many objects are loaded. You can clear MOE of loaded atoms with **MOE | File | Close**. To run QuaSAR-Descriptor, open the database in the Database Viewer and choose

DBV | Compute | Descriptors | Calculate

This command is grayed out if there is no molecule field in the database.

The following panel appears:



Database File Path and filename of the database into which molecular descriptors will be calculated. This field cannot be changed. To calculate descriptors for another database, re-issue the QuaSAR-Descriptor command for the new database.

Selected Entries Only Calculates descriptors on selected entries in the Database Viewer only.

Molecule Field Option menu listing all the molecular fields in the database. Choose the field containing the molecules for which descriptors will be calculated.

Auto Select The Database Fields, Selected Database Fields and Clear Selection buttons are fast ways to manipulate the selected descriptors in the list:

- **Database Fields** selects the descriptors that have codes identical to a field name of a numeric field in the database. Note: Descriptor codes already selected will remain selected.
- **Selected Database Fields** selects the descriptors that have codes identical to a field name of a selected numeric field in the Database Viewer. Note: Descriptor codes already selected will remain selected.

- **Clear Selection** deselects all descriptors in the list.

Descriptors Selected	Shows the number of descriptors selected for calculation. Note that, depending on the Class and Filter contents, not all available descriptors will be displayed in the list (see below). In other words, some descriptors may not be displayed even though they have been selected.
Class	The display of descriptors in the descriptor list may be restricted to a particular descriptor class by pressing on the corresponding Class button. The All button displays all classes. 2D and 3D descriptors are explained in the following sections.
Filter	<p>The contents of the Filter text field restricts the display of descriptors in the list. If no filter is specified, then no restriction other than the Class is made. If a filter is specified, then in addition to the restrictions imposed by Class, only those lines in the list with text containing each of the words in the Filter text field will be displayed. For example, if the Filter text is "don," the display is restricted to lines that contain the three consecutive letters (case insensitive) "don". If the Filter text is "don atom," the display is restricted to lines that contain the substring "don" as well as the substring "atom" (not necessarily in that order).</p> <p>The Apply button applies the filter to the list. Pressing <code>Enter</code> in the Filter text field will also apply the filter to the list. In general, it is not necessary to press <code>Enter</code> or Apply because the list will update automatically.</p> <p>The Clear button deletes all characters from the Filter text.</p>

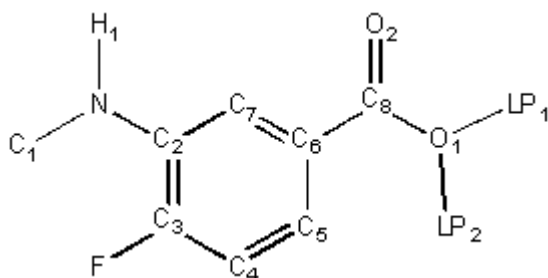
Select the molecular descriptors to calculate and press **OK**. The calculation may take some time depending on the nature of the descriptors and the size of the database. A log of the calculation will be printed to the SVL Commands window.

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].
density	Molecular mass density: Weight divided by vdw_vol ($\text{amu}/\text{\AA}^3$).
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 <code>mr</code> descriptor.
<code>h_mr</code>	Molar refractivity using a 4 parameter model based on Hueckel Theory [Labute 2015] with $r^2 = 0.99$, RMSE=0.20 on 1,947 molecules.
Weight	Molecular weight (including implicit hydrogens) in atomic mass units with atomic weights taken from [CRC 1994].
<code>h_logD</code>	The octanol/water distribution coefficient at pH 7.
<code>h_logP</code>	Log of the octanol/water partition coefficient using an 8 parameter model based on Hueckel Theory [Labute 2015] with $r^2 = 0.84$, RMSE=0.59 on 1,836 molecules. The calculated value is that of the protonation state of the structure as drawn.
<code>logP(o/w)</code>	Log of the octanol/water partition coefficient (including implicit hydrogens). This property is calculated from a ~100 parameter linear atom type model [LOGP 1998] with $r^2 = 0.931$, RMSE=0.393 on 1,827 molecules.
SlogP	Log of the octanol/water partition coefficient. 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 <code>logP(o/w)</code> descriptor. The training set for SlogP was ~7000 structures.
<code>logS</code>	Log of the aqueous solubility (mol/L). This property is calculated from an atom contribution linear atom type model [Hou 2004] with $r^2 = 0.90$, ~1,200 molecules.
<code>h_logS</code>	Log of the aqueous solubility (mol/L) using a 7 parameter model based on Hueckel Theory [Labute 2015] with $r^2 = 0.83$, RMSE=0.85 on 1,708 molecules. The calculated value is that of the protonation state of the structure as drawn.
mutagenic	Indicator of the presence of potentially toxic groups. A non-zero value indicates that the molecule contains a mutagenic group. The table of mutagenic groups is based on the Kazius set [Kazius 2005].
reactive	Indicator of the presence of reactive groups. A non-zero value indicates that the molecule contains a reactive group. The table of reactive groups is based on the Oprea set [Oprea 2000] and includes metals, phospho-, N/O/S-N/O/S

	single bonds, thiols, acyl halides, Michael Acceptors, azides, esters, etc.
rsynth	A value in [0,1] indicating the synthetic reasonableness, or feasibility, of the chemical structure. A value of 0 means it is unlikely that the molecule can be synthesized while a value of 1 means that it is likely that the molecule can be synthesized. The value reflects the fraction of heavy atoms in the molecule that can be traced back to starting materials fragments resulting from retrosynthetic disconnection rules. This molecular descriptor can also be calculated on molecules and databases using the SVL Retrosynth* function calls.
TPSA	Polar surface area (\AA^2) calculated using group contributions to approximate the polar surface area from connection table information only. The parameterization is that of Ertl <i>et al.</i> [Ertl 2000].
vdw_vol	van der Waals volume (\AA^3) calculated using a connection table approximation.
vdw_area	Area of van der Waals surface (\AA^2) calculated using a connection table approximation.

Hueckel Theory Descriptors

The Hueckel Theory descriptors are based on a modified Hueckel Theory calculation [Gerber 1998] that produces a sigma and pi charge for each atom and a fractional bond order (p- and d-orbitals). The calculation does not require coordinates and is based solely on the molecular topology. Consequently, the descriptors do not depend on the particular input resonance form and consistently treat electron withdrawal effects. **Note!** The Hueckel Theory descriptors report values based on the protonation state as given.

Code	Description
h_ema	Sum of hydrogen bond acceptor strengths [Gerber 1998].
h_emd	Sum of hydrogen bond donor strengths [Gerber 1998].
h_emd_C	Sum of hydrogen bond donor strengths of carbon atoms.
h_log_pbo	Sum of log (1 + pi bond order) for all bonds.
h_log_dbo	Sum of log (1 + d-hybrid bond order) for all bonds.
h_mr	Molar refractivity using a 4 parameter model based on Hueckel Theory [Labute 2015] with $r^2 = 0.99$, RMSE=0.20 on 1,947 molecules.

h_logP	Log of the octanol/water partition coefficient using an 8 parameter model based on Hueckel Theory [Labute 2015] with $r^2 = 0.84$, RMSE=0.59 on 1,836 molecules.
h_logS	Log of the aqueous solubility (mol/L) using a 7 parameter model based on Hueckel Theory [Labute 2015] with $r^2 = 0.83$, RMSE=0.85 on 1,708 molecules.

Another group of Hueckel Theory descriptors report quantities based on averages over all protonation states at pH 7. For a given input structure, the following procedure is used to enumerate the states.

1. Enumerate all potential $AH \rightarrow [A^-] + [H^+]$ reactions by protonating potential bases and deprotonating potential acids. Estimate a $pK_{i,j}$ value for each reaction between states i and j .
2. Identify all rings in the graph formed in the previous step and calculate a least squares correction to the $pK_{i,j}$ values such that the directed cycles sum to zero; i.e., minimize $|c|^2$ such that $\mathbf{R}(pK + c) = 0$ and replace the $pK_{i,j}$ with $pK_{i,j} + c$
3. Solve for the state log concentrations pC_i ; ensure mass balance in and out of each state and normalize the final concentrations (sum to 1). The output is a list of protonation states and log concentrations pC_i .

The above procedure is pH dependent and states are not considered if their concentrations are more than 6 log units from the most popular protonation state. The resulting states are then used to calculate the following descriptors:

Code	Description
h_pavgQ	The average total charge sum $\{ Q_i 10^{-pC_i} \}$ where Q_i is the total formal charge of state i .
h_pstates	The entropic count or fractional number of protonation states; 10^H where $H = \sum \{ pC_i 10^{-pC_i} \}$.
h_pstrain	The strain energy (kcal/mol) needed to convert all protonation states into the input protonation state: $(kT \ln 10) (pC_1 + \log \sum \{ 10^{-pC_i} \})$
h_pKa	The pKa of the reaction that removes a proton from the ensemble of states with a hydrogen count equal to the input structure; 14 is reported if there are no states with fewer hydrogens than the input.
h_pKb	The pKb of the reaction that adds a proton from the ensemble of states with a hydrogen count equal to the input structure; 14 is reported if there are no states with more hydrogens than the input.

h_logD	The octanol/water distribution coefficient at pH 7 calculated as a state average: $\log \sum \{ 10^{h_{\log P_i} - pC_i} \}$.
--------	--

Subdivided Surface Areas

The Subdivided Surface Areas are descriptors based on an approximate accessible van der Waals surface area (in \AA^2) calculation for each atom, v_i along with some other atomic property, p_i . The v_i are 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 calculated to be a_ICM times n .
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. Conjugated single bonds are not included (e.g. ester and peptide bonds).
b_1rotR	Fraction of rotatable single bonds: b_1rotN divided by b_heavy.
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 has order 1, is not in a ring, and has at least two heavy neighbors.
b_rotR	Fraction of rotatable bonds: b_rotN divided by b_heavy.
b_single	Number of single bonds (including implicit hydrogens). Aromatic bonds are not considered to be single bonds.
b_maxllen	Length of the longest single bond chain.
b_triple	Number of triple bonds. Aromatic bonds are not considered to be triple bonds.
chiral	The number of chiral centers.
chiral_u	The number of unconstrained chiral centers.
lip_acc	The number of O and N atoms.
lip_don	The number of OH and NH atoms.
lip_druglike	One if and only if lip_violation < 2 otherwise zero.
lip_violation	The number of violations of Lipinski's Rule of Five [Lipinski 1997].
nmol	The number of molecules (connected components).
opr_brigid	The number of rigid bonds from [Oprea 2000].
opr_leadlike	One if and only if opr_violation < 2 otherwise zero.
opr_nring	The number of ring bonds from [Oprea 2000].

opr_nrot	The number of rotatable bonds from [Oprea 2000].
opr_violation	The number of violations of Oprea's lead-like test [Oprea 2000].
rings	The number of rings.
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 heavy atom degree d_i (number of heavy neighbors) and v_i . 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 p_3 the number of paths of length 3.

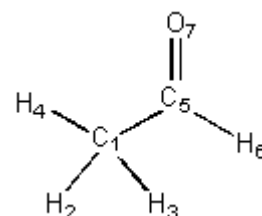
Code	Description
chi0	Atomic connectivity index (order 0) from [Hall 1991] and [Hall 1977]. This is calculated as the sum of $1/\sqrt{d_i}$ over all heavy atoms i with $d_i > 0$.
chi0_C	Carbon connectivity index (order 0). This is calculated as the sum of $1/\sqrt{d_i}$ over all carbon atoms i with $d_i > 0$.
chi1	Atomic connectivity index (order 1) from [Hall 1991] and [Hall 1977]. 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 1977]. 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 1977]. 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: $(s-1)(s-3)^2 / p_3^2$ for odd n , and $(s-3)(s-2)^2 / p_3^2$ 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	1	0	0	0	1	1	1	1	2	2	
H2	1	0	0	0	0	0	0	0	1	0	2	2	2	3	3
H3	1	0	0	0	0	0	0	0	1	2	0	2	2	3	3
H4	1	0	0	0	0	0	0	0	1	2	2	0	2	3	3
C5	1	0	0	0	0	1	1	1	1	2	2	2	0	1	1
H6	0	0	0	0	1	0	0	0	2	3	3	3	1	0	2
O7	0	0	0	0	1	0	0	0	2	3	3	3	1	2	0



Petitjean [Petitjean 1992] defines the **eccentricity** of a vertex to be the longest path from that vertex to any other vertex in the graph. The graph **radius** is the smallest vertex eccentricity in the graph and the graph **diameter** as the largest vertex eccentricity. These values are calculated using the distance matrix and are used for several descriptors described below.

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].
BCUT_PEOE_0 BCUT_PEOE_1 BCUT_PEOE_2 BCUT_PEOE_3	The BCUT descriptors [Pearlman 1998] are calculated from the eigenvalues of a modified adjacency matrix. Each ij entry of the adjacency matrix takes the value $1/\sqrt{b_{ij}}$ where b_{ij} is the formal bond order between bonded atoms i and j . The diagonal takes the value of the PEOE partial charges. The resulting eigenvalues are sorted and the smallest, 1/3-ile, 2/3-ile and largest eigenvalues are reported.
BCUT_SLOGP_0 BCUT_SLOGP_1 BCUT_SLOGP_2 BCUT_SLOGP_3	The BCUT descriptors using atomic contribution to logP (using the Wildman and Crippen SlogP method) instead of partial charge.
BCUT_SMR_0 BCUT_SMR_1 BCUT_SMR_2 BCUT_SMR_3	The BCUT descriptors using atomic contribution to molar refractivity (using the Wildman and Crippen SMR method) instead of partial charge.
diameter	Largest value in the distance matrix [Petitjean 1992].
petitjean	Value of (diameter - radius) / diameter.

GCUT_PEOE_0 GCUT_PEOE_1 GCUT_PEOE_2 GCUT_PEOE_3	The GCUT descriptors are calculated from the eigenvalues of a modified graph distance adjacency matrix. Each ij entry of the adjacency matrix takes the value $1/\sqrt{d_{ij}}$ where d_{ij} is the (modified) graph distance between atoms i and j . The diagonal takes the value of the PEOE partial charges. The resulting eigenvalues are sorted and the smallest, 1/3-ile, 2/3-ile and largest eigenvalues are reported.
GCUT_SLOGP_0 GCUT_SLOGP_1 GCUT_SLOGP_2 GCUT_SLOGP_3	The GCUT descriptors using atomic contribution to logP (using the Wildman and Crippen SlogP method) instead of partial charge.
GCUT_SMR_0 GCUT_SMR_1 GCUT_SMR_2 GCUT_SMR_3	The GCUT descriptors using atomic contribution to molar refractivity (using the Wildman and Crippen SMR method) instead of partial charge.
petitjeanSC	Petitjean graph Shape Coefficient as defined in [Petitjean 1992]: $(\text{diameter} - \text{radius}) / \text{radius}$.
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 .
wienerPath	Wiener path number: half the sum of all the distance matrix entries as defined in [Balaban 1979] and [Wiener 1947].
wienerPol	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. That is, hydrogens are suppressed during the calculation. The atom typing mechanism is located in the file `$MOE/lib/svl/ph4.svl/ph4type.svl` which is a rule-based system for assigning pharmacophore features to atoms. 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_donacc	Number of hydrogen bond donor plus number of hydrogen bond acceptor atoms.
a_hyd	Number of hydrophobic atoms.
vsa_acc	Approximation to the sum of VDW surface areas (\AA^2) 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 (\AA^2).
vsa_base	Approximation to the sum of VDW surface areas of basic atoms (\AA^2).
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) (\AA^2).
vsa_hyd	Approximation to the sum of VDW surface areas of hydrophobic atoms (\AA^2).
vsa_other	Approximation to the sum of VDW surface areas (\AA^2) of atoms typed as "other".
vsa_pol	Approximation to the sum of VDW surface areas (\AA^2) of polar atoms (atoms that are both hydrogen bond donors and acceptors), 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. Electronegativity values are determined by parameterization found in the SVL source code file

\$MOE/lib/svl/calc.svl/charge.svl. 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).

Partial charges from forcefields can be used by energy minimizing the database structures (which will store the charges), then using the Q_ variant of the descriptors.

Warning! Most partial charge methods require that all hydrogens be explicit in the stored structures. This can be a source of error when recalculating descriptors on novel structures during QSAR model evaluation.

Let q_i denote the partial charge of atom i as defined above. Let v_i be the van der Waals surface area (\AA^2) 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_PRC-	Relative negative partial charge: the smallest negative q_i divided by the sum of the negative q_i . Q_PRC- is identical to PRC- 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.

Protein Property Descriptors

Various physicochemical protein properties can be calculated. These are described under the application [Protein Properties](#).

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.

Note: All the 3D descriptors operate on structures found in the database as is; that is, no hydrogens are added or removed. Furthermore, most descriptors assume that partial charges are stored with the structures in the database.

Potential Energy Descriptors

The energy descriptors use the MOE potential energy model to calculate energetic quantities (in kcal/mol) from stored 3D conformations. The currently loaded forcefield is used along with the current potential setup as defined in the [Potential Setup](#) panel. 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. To use the potential energy descriptors:

1. Open **MOE | Footer | forcefield** to open the Potential Setup panel and select an appropriate forcefield.
2. Decide on which weights should be set to 0; for example, if the solvation model is not required, set its weight to 0. It is not sufficient to disable the term with the Enable/Disable checkbox. Terms with zero weights will not play a role in any of the descriptor calculations. See [pot_Setup](#) for more details.
3. Decide on which terms should be enabled. A number of energy descriptors will honor the state of the enabled flags (see below).

One remaining issue is that of partial charges. The internal coordinate energy descriptors assume that if the partial charges stored in the database are essentially zero, then the potential model charges will be calculated prior to calculating the energy. If the partial charges are not essentially zero, then they are left as is for the descriptor calculation. In this way, partial charges calculated with an external program will be left undisturbed whereas molecules with no charges will be given those recommended by the forcefield.

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 (Bonded) 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 bonded terms disabled. 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 (Bonded) flag is ignored, but the term weight is applied.
E_sol	Solvation energy. In the Potential Setup panel, the term enable parameter (Solvation menu) 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 (Bonded) flag is ignored, but the term weight is applied.
E_str	Bond stretch potential energy. In the Potential Setup panel, the term enable (Bonded) 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 E descriptor. The local minimum energy is the value of the E 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 (Bonded) 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. 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 on the currently loaded atoms.
E_rnb	Non-bonded interaction energy (external reference frame: x3d) between the stored molecule and a "receptor" currently

	loaded in MOE. It is similar to the other E_r* calls in that it is an interaction energy term.
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 $G - L - R$. 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 on the currently loaded atoms.
E_rvdw	van der Waals interaction energy (external reference frame: x3d) between the stored molecule and the atoms currently loaded. The term enable flag is ignored, but the term weight is applied.

MOPAC Descriptors

The MOPAC [MOPAC] descriptors are calculated by the version of [MOPAC](#) distributed with MOE.

Code	Description
AM1_dipole	The dipole moment calculated using the AM1 Hamiltonian [MOPAC].
AM1_E	The total SCF energy (kcal/mol) calculated using the AM1 Hamiltonian [MOPAC].
AM1_Eele	The electronic energy (kcal/mol) calculated using the AM1 Hamiltonian [MOPAC].
AM1_HF	The heat of formation (kcal/mol) calculated using the AM1 Hamiltonian [MOPAC].
AM1_IP	The ionization potential (kcal/mol) calculated using the AM1 Hamiltonian [MOPAC].
AM1_LUMO	The energy (eV) of the Lowest Unoccupied Molecular Orbital calculated using the AM1 Hamiltonian [MOPAC].
AM1_HOMO	The energy (eV) of the Highest Occupied Molecular Orbital calculated using the AM1 Hamiltonian [MOPAC].
MNDO_dipole	The dipole moment calculated using the MNDO Hamiltonian [MOPAC].
MNDO_E	The total SCF energy (kcal/mol) calculated using the MNDO Hamiltonian [MOPAC].
MNDO_Eele	The electronic energy (kcal/mol) calculated using the MNDO Hamiltonian [MOPAC].

MNDO_HF	The heat of formation (kcal/mol) calculated using the MNDO Hamiltonian [MOPAC].
MNDO_IP	The ionization potential (kcal/mol) calculated using the MNDO Hamiltonian [MOPAC].
MNDO_LUMO	The energy (eV) of the Lowest Unoccupied Molecular Orbital calculated using the MNDO Hamiltonian [MOPAC].
MNDO_HOMO	The energy (eV) of the Highest Occupied Molecular Orbital calculated using the MNDO Hamiltonian [MOPAC].
PM3_dipole	The dipole moment calculated using the PM3 Hamiltonian [MOPAC].
PM3_E	The total SCF energy (kcal/mol) calculated using the PM3 Hamiltonian [MOPAC].
PM3_Eele	The electronic energy (kcal/mol) calculated using the PM3 Hamiltonian [MOPAC].
PM3_HF	The heat of formation (kcal/mol) calculated using the PM3 Hamiltonian [MOPAC].
PM3_IP	The ionization potential (kcal/mol) calculated using the PM3 Hamiltonian [MOPAC].
PM3_LUMO	The energy (eV) of the Lowest Unoccupied Molecular Orbital calculated using the PM3 Hamiltonian [MOPAC].
PM3_HOMO	The energy (eV) of the Highest Occupied Molecular Orbital calculated using the PM3 Hamiltonian [MOPAC].

Surface Area, Volume and Shape Descriptors

The following descriptors depend on the structure connectivity and conformation (dimensions are measured in Å). The `vsurf_` descriptors are similar to the VolSurf descriptors [Cruciani 2000]; these descriptors have been shown to be useful in pharmacokinetic property prediction. The `pmi1`, `pmi2`, and `pmi3` descriptors are calculated after [Sauer 2003].

Note: The principal moment of inertia and related calculations are performed in units of daltons (AMU) and angstroms.

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>vol</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).
pmi1	First diagonal element of diagonalized moment of inertia tensor.
pmi2	Second diagonal element of diagonalized moment of inertia tensor.
pmi3	Third diagonal element of diagonalized moment of inertia tensor.
npr1	Normalized PMI ratio $\text{pmi1}/\text{pmi3}$.
npr2	Normalized PMI ratio $\text{pmi2}/\text{pmi3}$.
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.
vsurf_V	Interaction field volume
vsurf_S	Interaction field surface area

vsurf_R	Surface rugosity
vsurf_G	Surface globularity
vsurf_W*	Hydrophilic volume (8 descriptors)
vsurf_IW*	Hydrophilic integy moment (8 descriptors)
vsurf_CW*	Capacity factor (8 descriptors)
vsurf_EWmin*	Lowest hydrophilic energy (3 descriptors)
vsurf_DW*	Contact distances of vsurf_EWmin (3 descriptors)
vsurf_D*	Hydrophobic volume (8 descriptors)
vsurf_ID*	Hydrophobic integy moment (8 descriptors)
vsurf_EDmin*	Lowest hydrophobic energy (3 descriptors)
vsurf_DD*	Contact distances of vsurf_DDmin (3 descriptors)
vsurf_HL*	Hydrophilic-Lipophilic (2 descriptors)
vsurf_A	Amphiphilic moment
vsurf_CP	Critical packing parameter
vsurf_Wp*	Polar volume (8 descriptors)
vsurf_HB*	H-bond donor capacity (8 descriptors)

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 (in Å²) 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 Calculate Descriptors conventions described below.
2. Load the descriptor module into MOE with `load'mydesc.svl'`. Note that if the **Calculate Descriptors** 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):

```
#svl
//      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 = mol_Create 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 line containing #svl, then a comment header, followed by SVL compiler directives:

```
#svl
//      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 Tasks window, and gives 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 **Calculate Descriptors** 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_calc_MyDescriptors`. The association is created by using 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 [mol_Extract](#).

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.

SVL Commands

[QuaSAR_DescriptorMDB](#)
[QuaSAR_Descriptor_List](#)
[DescriptorExprSyntax](#)
[DescriptorExprCodes](#)
[DescriptorExprEval](#)

References

- [Wiener 1947] Wiener, H.; Structural Determination of Paraffin Boiling Points; *Journal of the American Chemical Society* 69 (**1947**) 17–20.
- [Hall 1977] Hall, L.H., Kier, L.B.; The Nature of Structure-Activity Relationships and Their Relation to Molecular Connectivity; *Eur. J. Med. Chem* 12 (**1977**) 307.
- [Balaban 1979] Balaban, A.T.; Five New Topological Indices for the Branching of Tree-Like Graphs; *Theoretica Chimica Acta* 53 (**1979**) 355–375.
- [Gasteiger 1980] Gasteiger, J., Marsili, M.; Iterative Partial Equalization of Orbital Electronegativity - A Rapid Access to Atomic Charges; *Tetrahedron* 36 (**1980**) 3219.
- [Balaban 1982] Balaban, A.T.; Highly Discriminating Distance-Based Topological Index; *Chemical Physics Letters* 89 No. 5 (**1982**) 399–404.
- [Stanton 1990] Stanton, D., Jurs, P.; Development and Use of Charged Partial Surface-Area Structural Descriptors in Computer-Assisted Quantitative Structure-Property Relationship Studies; *Anal. Chem.* 62 (**1990**) 2323–2329.
- [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* 2 (**1991**).
- [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 (**1992**) 331–337.
- [MOPAC] Stewart, J.J.P.; *MOPAC Manual (Seventh Edition)*; (**1993**).
- [CRC 1994] *CRC Handbook of Chemistry and Physics*. CRC Press (**1994**).
- [Lipinski 1997] Lipinski, C.A., Lombardo, F., Dominy, B.W. and Feeney, P.J.; Experimental and Computational Approaches to Estimate

Solubility and Permeability in Drug Discovery and Development Settings; *Adv. Drug Deliv. Rev.* 23 (**1997**) 3–25.

- [MREF 1998] Labute, P.; MOE Molar Refractivity Model *unpublished*. Source code in \$MOE/lib/avl/quasar.avl/q_mref.avl (**1998**).
- [Pearlman 1998] Pearlman, R.S., Smith, K.M.; Novel Software Tools for Chemical Diversity; *Persp. Drug. Disc. Des.* 9/10/11 (**1998**) 339–353.
- [Gerber 1998] Gerber, P.R.; Charge distribution from a simple molecular orbital type calculation and non-bonding interaction terms in the force field MAB; *J. Comput.-Aid. Mol. Des.* 12 (**1998**) 37–51.
- [LOGP 1998] Labute, P.; MOE LogP(Octanol/Water) Model *unpublished*. Source code in \$MOE/lib/avl/quasar.avl/q_logp.avl (**1998**).
- [Crippen 1999] Wildman, S.A., Crippen, G.M.; Prediction of Physiochemical Parameters by Atomic Contributions; *J. Chem. Inf. Comput. Sci.* 39 No. 5 (**1999**) 868–873.
- [Cruciani 2000] Cruciani, G., Crivori, P., Carrupt, P.-A., Testa, B.; Molecular Fields in Quantitative Structure-Permeation Relationships: the VolSurf Approach; *J. Mol. Struct. (Theochem)* 503 (**2000**) 17–30.
- [Labute 2000] Labute, P.; A Widely Applicable Set of Descriptors; *Journal of Molecular Graphics and Modelling* 18 (**2000**) 464–477.
- [Ertl 2000] Ertl, P., Rohde, B., Selzer, P.; Fast Calculation of Molecular Polar Surface Area as a Sum of Fragment-Based Contributions and Its Application to the Prediction of Drug Transport Properties; *J. Med. Chem.* 43 (**2000**) 3714–3717.
- [Oprea 2000] Oprea, Tudor I.; Property Distribution of Drug-Related Chemical Databases; *J. Comp. Aid. Mol. Des.* 14 (**2000**) 251–264.
- [Sauer 2003] Sauer, W.H.B. and Schwarz, M.K.; Molecular Shape Diversity of Combinatorial Libraries: A Prerequisite for Broad Bioactivity; *J. Chem. Inf. Comput. Sci.* 43, (**2003**) 987–1003.
- [Hou 2004] Hou, T.J., Xia, K., Zhang, W., Xu, X.J.; ADME Evaluation in Drug Discovery. 4. Prediction of Aqueous Solubility Based on Atom Contribution Approach; *J. Chem. Inf. Comput. Sci.* 44 (**2004**) 266–275.
- [Kazius 2005] Kazius, J., McGuire, R., Bursi, R.; Derivation and Validation of Toxicophores for Mutagenicity Prediction; *J. Med. Chem.* 48 (**2005**) 312–320.
- [Labute 2015] Labute, P.; MOE h_mr, h_logP and h_logS Models *unpublished*. Source code in \$MOE/lib/avl/quasar.avl/q_eht.avl (**2015**).

See Also

[Synthetic Feasibility Functions](#)

[QuaSAR-Model](#)

[QuaSAR Overview](#)

[QuaSAR Tutorial](#)

[MOE Table of Contents](#)



Copyright © 1997–2015 [Chemical Computing Group Inc.](#)

info@chemcomp.com