3D descriptors in RDKit

Guillaume GODIN

Principal Scientist

Chemoinformatic & Data science

Agenda

- Docker installation
- List of new descriptors available in RDKit
- Review of the new descriptors
  - Definition
  - Array names
  - Script examples
  - Few application in modelling
- Conclusion

Docker Installation with « eigen3 »

- Docker on mac osx thanks to Tim Dudgeon (https://hub.docker.com/r/informaticsmatters/rdkit)
- Two little changes in the Dockerfile to have 3D descriptors:
  - **RUN** apt-get install libeigen3-dev -y
  - **RUN** git clone https://github.com/rdkit/rdkit.git
- Installation method:
  - Install Docker (https://www.docker.com)
  - Copy Tim « Dockerfile » and apply those modifications
  - docker build .
  - docker run -it --rm e55a59553eda bash (where « e55a59553eda » is your Docker image id)

List of new 3D descriptors available in RDKit

Those descriptors where created in order to encode 3D molecular information related to atomic properties and/or geometric properties

All of them can be used in QSAR models

Current implementation use Eigen 3 c++ a fast library for matrix operations, singular value decomposition, etc…

Name	Number	Date
RDF	210	1999
MORSE	224	1996
AUTOCORR3D	80	1984
AUTOCORR2D	192	1984
GETAWAY	273	2002
WHIM	114	1994/1998
Total	**1093**	

Descriptors implemented in opensource tools in 2015

Type of descriptors	Dimension	Number of descriptors	The origin of features
Constitutional descriptors	1	309	A, B, C, D, E, F
Molecular format descriptors	1	6	A
Autocorrelation descriptors	2	467	C, B, E, F
Basak descriptors	2	63	B, E
BCUT descriptors	2	12	C, E
Burden descriptors	2	160	B, E
Connectivity descriptors	2	194	C, B, D, E, F
E-state descriptors	2	734	B, E
Kappa descriptors	2	92	C, B, E
Molecular property descriptors	2	55	A, B, C, D, E, F
Quantum chemical descriptors	2	7	C, E
Topological descriptors	2	376	B, C, D, E, F
MOE-type descriptors	2	118	B, D
Charge descriptors	2	25	B
3D Autocorrelation descriptors	3	80	E
CPSA descriptors	3	116	B, C, E, F
RDF descriptors	3	390	B, E
Geometrical descriptors	3	62	B, C, E, F
MoRSE descriptors	3	210	B
WHIM descriptors	3	195	B, C, E, F

A, B, C, D, E, F stands for Pybel, Chemopy, CDK, RDKit, PaDEL, and BlueDesc, respectively

Jie Dong, Dong-Sheng Cao, Hong-Yu Miao, Shao Liu, Bai-Chuan Deng, Yong-Huan Yun, Ning-Ning Wang, Ai-Ping Lu, Wen-Bin Zeng, Alex F. Chen J Cheminform. 2015; 7: 60. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4674923/

RDKit RDF descriptors

The general Radial Distribution Function (RDF) is an expression for the probability distribution of distances r_{ij} between two points i and j within a three-dimensional space of N points:

$$g(r) = \sum_{i}^{N-1} \sum_{j>i}^{N} e^{-B(r-r_{ij})^2} \qquad (16)$$

The exponential term leads to a Gaussian distribution around the distance r_{ij} with a half-peak width depending on the smoothing parameter B.

B=100 similar to Dragon

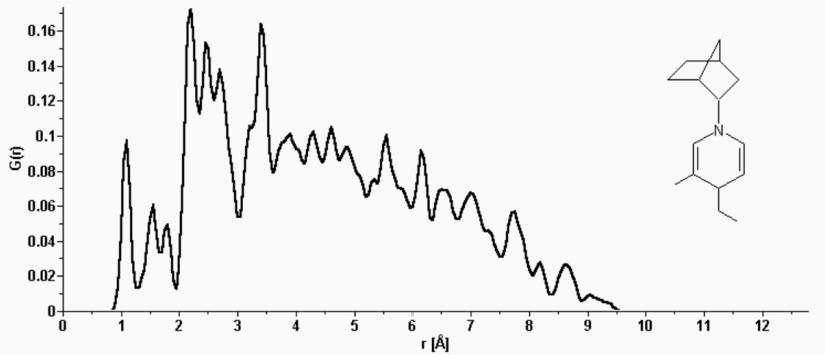


Figure 3. RDF descriptor calculated for a polycyclic system according to equation (16). The complete RDF function represents a probability distribution; the individual peaks are related to the relative frequencies of atom distances in the molecule.

Hemmer, Steinhauer, Gasteiger, J. Vibrat. Spect., (1999), 19, 151 -164.
https://opus4.kobv.de/opus4-fau/files/549/MarkusHemmerDissertation.pdf

To go futher

- It is independent of the number of atoms, i.e. the size of a molecule
- It is unique regarding the three-dimensional arrangement of the atoms
- It is invariant against translation and rotation of the entire molecule.
- RDF code can be restricted to specific atom types or distance ranges to represent specific information in a certain three-dimensional structure space, e.g. to describe steric hindrance or the structure and/or activity properties of a molecule.
- RDF code is interpretable by using simple rule sets, and thus provides the possibility of converting the code back into the corresponding 3D structure.
- Besides information about interatomic distances in the entire molecule, the RDF code provides further valuable information, e.g. about bond distances, ring types, planar and non-planar systems and atom types.

http://michem.disat.unimib.it/chm/download/materiale/geometrical_descriptors.pdf

RDF Vector output definitions (size: 210)

Seven groups (for 6 atomic properties + unweighted):

Position	Code	Name
1-30	RDFu*	Unweighted
31-60	RDFm	Relative atomic Mass
61-90	RDFv	Relative van der Waals volume
91-120	RDFe	Relative Electronegativity
121-150	RDFp	Relative atomic polarizability
151-180	RDFi	Relative atomic ion polarity
181-210	RDFs	Relative IState

Thirty radial distances for each groups

*RDF010x RDF015x RDF020x RDF025x RDF030x RDF035x RDF040x RDF045x RDF050x RDF055x RDF060x RDF065x RDF070x RDF075x RDF080x RDF085x RDF090x RDF095x RDF100x RDF105x RDF110x RDF115x RDF120x RDF125x RDF130x RDF135x RDF140x RDF145x RDF150x RDF155x

Example of Scripts

```
from __future__ import print_function
from rdkit import Chem
from rdkit.Chem import rdMolDescriptors as rdMD
from rdkit.Chem import AllChem
m = Chem.MolFromSmiles('Cc1ccccc1')
m2 =Chem.AddHs(m)
AllChem.EmbedMolecule(m2,AllChem.ETKDG())
m2 = Chem.RemoveHs(m2)
r=rdMD.CalcRDF(m)
print(r)
r=rdMD.CalcRDF(m2)
print(r)
```

```
>>> print (r)
[0.0, 2.154, 0.0, 2.98, 0.004, 0.006, 0.009, 0.001, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0
, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 2.1
54, 0.0, 2.98, 0.004, 0.006, 0.009, 0.001, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 2.154, 0.0,
2.98, 0.004, 0.006, 0.009, 0.001, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0
.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 2.154, 0.0, 2.98, 0.0
04, 0.006, 0.009, 0.001, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 2.154, 0.0, 2.98, 0.004, 0.006
, 0.009, 0.001, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 2.154, 0.0, 2.98, 0.004, 0.006, 0.009,
0.001, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0
, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 16.379, 0.0, 22.462, 0.05, 0.023, 0.035, 0.003,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0]
```

```
>>> r=rdMD.CalcRDF(m2)
>>> print(r)
[0.0, 2.436, 0.0, 3.993, 0.033, 0.001, 0.019, 0.018, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 2.
436, 0.0, 3.993, 0.033, 0.001, 0.019, 0.018, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 2.436, 0.0
, 3.993, 0.033, 0.001, 0.019, 0.018, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0
, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 2.436, 0.0, 3.993,
 0.033, 0.001, 0.019, 0.018, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0
.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 2.436, 0.0, 3.993, 0.033,
0.001, 0.019, 0.018, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 2.436, 0.0, 3.993, 0.033, 0.001, 0
.019, 0.018, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 20.561, 0.0, 30.777, 0.24, 0.003, 0.077, 0
.072, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
```

Some Applications of RDF:

- **Radial Distribution Function descriptors for predicting affinity for vitamin D receptor (https://www.ncbi.nlm.nih.gov/pubmed/18068275 Gonzalez, Gandara, Fall, Gomez** Eur J Med Chem. 2008 Jul;43(7):1360-5**)**
- **Recent Trends on QSAR in the Pharmaceutical Perceptions (**Tareq, Khan, Bentham e-books 2012)
- **Using Infrared spectrum database & neural network (Kohonen SOM) model to determine correct 3D structures (Expert Systems in Chemistry Research,** Hemmer, 2007)

RDKit 3D-MORSE

3D Molecule Representation of Structure based on Electron diffraction

Similarly to RDF whose descriptors were introduced in 1996 by Schuur, Selzer, Gasteiger
with the motivation for encoding 3D structure of a molecule by a fixed number of variables

$$I(s) = \sum_{i=1}^{A-1} \sum_{j=i+1}^{A} w_i \cdot w_j \cdot \frac{\sin(s \cdot r_{ij})}{s \cdot r_{ij}}$$

S : 0, …, 31 & rij is the distance between atoms i & j

Another implementation in c++ is available using 3D structure from MOPAC: https://github.com/devinyak/3dmorse
(Devinyak, Havrylyuk, Lesyk, J. Mol. Graph. Model., 2014.)

Schuur, J. & Gasteiger, J.,Software Development in Chemistry - Vol. 10 (Gasteiger, J., ed.), Fachgruppe Chemie-lnformation-Computer (CIC),Frankfurt am Main, (1996).

3D-MORSE Vector output definition (size: 224)

Seven groups (for 6 atomic properties + unweighted)

Position	Code	Name
1-32	Moru*	Unweighted
33-64	Morm	Relative atomic Mass
65-96	Morv	Relative van der Waals volume
97-128	More	Relative Electronegativity
129-160	Morp	Relative atomic polarizability
161-192	Mori	Relative atomic ion polarity
193-224	Mors	Relative IState

Thirty-two morses for each groups

*Mor01x Mor02x Mor03x Mor04x Mor05x Mor06x Mor07x Mor08x Mor09uMor10x Mor11x Mor12x Mor13x Mor14x Mor15x Mor16x Mor17x Mor18uMor19x Mor20x Mor21x Mor22x Mor23x Mor24x Mor25x Mor26x Mor27uMor28x Mor29x Mor30x Mor31x Mor32x

Example of Scripts

```
from __future__ import print_function
from rdkit import Chem
from rdkit.Chem import rdMolDescriptors as rdMD
from rdkit.Chem import AllChem
m = Chem.MolFromSmiles('Cc1ccccc1')
m2 = Chem.AddHs(m)
AllChem.EmbedMolecule(m2,AllChem.ETKDG())
m2 = Chem.RemoveHs(m2)
rdMD.CalcMORSE(m)
```

```
>>> rdMD.CalcMORSE(m)
[21.0, 7.124, -0.794, -0.37, -1.217, 0.463, 1.089, -0.493, -0.505, 0.079, 0.217, 0.33
6, -0.25, -0.45, 0.388, 0.082, 0.027, -0.139, -0.247, 0.35, 0.009, -0.069, -0.094, -0
.075, 0.224, 0.02, -0.146, -0.04, 0.05, 0.076, 0.057, -0.182, 21.0, 7.124, -0.794, -0
.37, -1.217, 0.463, 1.089, -0.493, -0.505, 0.079, 0.217, 0.336, -0.25, -0.45, 0.388,
0.082, 0.027, -0.139, -0.247, 0.35, 0.009, -0.069, -0.094, -0.075, 0.224, 0.02, -0.14
6, -0.04, 0.05, 0.076, 0.057, -0.182, 21.0, 7.124, -0.794, -0.37, -1.217, 0.463, 1.08
9, -0.493, -0.505, 0.079, 0.217, 0.336, -0.25, -0.45, 0.388, 0.082, 0.027, -0.139, -0
.247, 0.35, 0.009, -0.069, -0.094, -0.075, 0.224, 0.02, -0.146, -0.04, 0.05, 0.076, 0
.057, -0.182, 21.0, 7.124, -0.794, -0.37, -1.217, 0.463, 1.089, -0.493, -0.505, 0.079
, 0.217, 0.336, -0.25, -0.45, 0.388, 0.082, 0.027, -0.139, -0.247, 0.35, 0.009, -0.06
9, -0.094, -0.075, 0.224, 0.02, -0.146, -0.04, 0.05, 0.076, 0.057, -0.182, 21.0, 7.12
4, -0.794, -0.37, -1.217, 0.463, 1.089, -0.493, -0.505, 0.079, 0.217, 0.336, -0.25, -
0.45, 0.388, 0.082, 0.027, -0.139, -0.247, 0.35, 0.009, -0.069, -0.094, -0.075, 0.224
, 0.02, -0.146, -0.04, 0.05, 0.076, 0.057, -0.182, 21.0, 7.124, -0.794, -0.37, -1.217
, 0.463, 1.089, -0.493, -0.505, 0.079, 0.217, 0.336, -0.25, -0.45, 0.388, 0.082, 0.02
7, -0.139, -0.247, 0.35, 0.009, -0.069, -0.094, -0.075, 0.224, 0.02, -0.146, -0.04, 0
.05, 0.076, 0.057, -0.182, 215.007, 84.572, -10.413, -3.097, -14.519, 4.553, 12.938,
-4.715, -6.332, 0.175, 2.793, 4.348, -2.562, -6.031, 4.302, 1.5, 0.62, -1.787, -3.659
, 4.463, 0.791, -1.213, -1.394, -0.952, 2.904, 0.695, -2.334, -0.704, 1.061, 0.972, 0
.676, -2.608]
```

Some Applications of 3D-MORSE:

- Prediction of rate constants for radical degradation of aromatic pollutants in water matrix
- Infrared spectra simulation of substituted benzene derivatives on the basis of a 3d structure representation
- Theoretical prediction of antiproliferative activity against murine leukemia tumor cell line (L1210)
- QSAR studies about cytotoxicity of benzophenazines with dual inhibition toward both topoisomerases i and ii
- QSAR analysis for heterocyclic antifungals
- QSAR study of carboxylic acid derivatives as HIV-1 integrase inhibitors.
- Prediction of cytotoxicity data (CC50) of anti-HIV 5-pheny-l-phenylamino-1H-imidazole derivatives by artificial neural network trained with Levenberg–Marquardt algorithm
- Docking and quantitative structure– activity relationship studies for sulfonyl hydrazides as inhibitors of cytosolic human branched-chain amino acid aminotransferase
- Simultaneously optimized support vector regression combined with genetic algorithm for QSAR analysis of KDR/VEGFR-2 inhibitors
- QSAR, QSPR and QSRR in terms of 3D-MoRSE descriptors for in silico screening of clofibric acid analogues
- LogP, MR, polarisability of Phenols derivates
- ….

References in: Devinyak, Havrylyuk, Lesyk, J. Mol. Graph. Model., 2014. DOI: 10.1016/j.jmgm.2014.10.006

RDKit AUTOCORR3D:

- Autocorrelation means how « similar » are two

Signals separate between a specific distance

$$AC_l = \sum_{i=1}^{n-l} f(x_i) \cdot f(x_{i+l})$$

Example of Scripts

```
from __future__ import print_function

from rdkit import Chem

from rdkit.Chem import rdMolDescriptors as rdMD

from rdkit.Chem import AllChem

m = Chem.MolFromSmiles('Cc1ccccc1')

m2 = Chem.AddHs(m)
AllChem.EmbedMolecule(m2,AllChem.ETKDG())

m2 = Chem.RemoveHs(m2)

rdMD.CalcAUTOCORR3D(m)
```

m2 is a 3D molecule

```
>>> rdMD.CalcAUTOCORR3D(m)
[05:42:28]

****
Pre-condition Violation
molecule has no conformers
Violation occurred on line 178 in file /rdkit/Code/GraphMol/Descripto
rs/AUTOCORR3D.cpp
Failed Expression: mol.getNumConformers() >= 1
****

Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
RuntimeError: Pre-condition Violation
        molecule has no conformers
        Violation occurred on line 178 in file Code/GraphMol/Descript
ors/AUTOCORR3D.cpp
        Failed Expression: mol.getNumConformers() >= 1
        RDKIT: 2017.09.1.dev1
        BOOST: 1_62

>>> rdMD.CalcAUTOCORR3D(m2)
[0.177, 0.501, 0.79, 0.892, 0.571, 0.167, 0.0, 0.0, 0.0, 0.0, 0.1, 0.
207, 0.197, 0.086, 0.018, 0.001, 0.0, 0.0, 0.0, 0.0, 0.115, 0.258, 0.
301, 0.2, 0.076, 0.012, 0.0, 0.0, 0.0, 0.0, 0.172, 0.48, 0.747, 0.824
, 0.516, 0.148, 0.0, 0.0, 0.0, 0.0, 0.125, 0.293, 0.372, 0.285, 0.127
, 0.024, 0.0, 0.0, 0.0, 0.0, 0.194, 0.58, 0.946, 1.153, 0.785, 0.243,
 0.0, 0.0, 0.0, 0.0, 0.513, 1.255, 1.747, 1.516, 0.758, 0.167, 0.0, 0
.0, 0.0, 0.0, 0.127, 0.301, 0.388, 0.307, 0.14, 0.028, 0.0, 0.0, 0.0,
 0.0]
```

You must to have a 3D molecule….

AUTOCORR3D Vector output definition (size: 80)

Eight groups (for 7 atomic properties + unweighted):

Position	Code	Name
1-10	TDBu*	Unweighted
11-20	TDBm	Relative atomic Mass
21-30	TDBv	Relative van der Waals volume
31-40	TDBe	Relative Electronegativity
41-50	TDBp	Relative atomic polarizability
51-60	TDBi	Relative atomic ion polarity
61-70	TDBs	Relative IState
71-80	TDBr	Relative cor ?

Ten AC for each groups

*TDB01x TDB02x TDB03x TDB04x TDB05x TDB06x TDB07x TDB08x TDB09x TDB10x

Some Applications of AUTOCORR3D:

- QSAR of inhibition by Flavonoid derivatives of p56lck Protein Tyrosine Kinase (PTK) (2006)
- Quantitative Structure–Activity Relationship (QSAR) Approximation for Cadmium Oxide (CdO) and Rhodium (III) Oxide (Rh2 O3) Nanoparticles as Anti-Cancer Drugs for the Catalytic Formation of Proviral DNA from Viral RNA Using Multiple Linear and Non-Linear Correlation Approach (2016)
- Similarity based classification studies for prediction of ABCB1 (P-glycoprotein) substrates and non-substrates (2013)

https://bib.irb.hr/prikazi-rad?&rad=272960

http://www.aclr.com.es/clinical-research/quantitative-structureactivity-relationship-qsar-approximation-for-cadmium-oxide-cdo-and-rhodium-iii-oxide-rh2o3-nanoparticles-as.pdf

http://othes.univie.ac.at/29514/1/2013-05-25_9815835.pdf

RDKit AUTOCORR2D

- Not 3D descriptor vector but it's very easy to implement it when you already have AUTOCORR3D
- So I add them as an extrat bonus ;-)
- We implement 4 AC methods in RDKit:
  - Spatial autocorrelation of a Topological Structure (ATS) aka « **Moreau Broto »**
  - Average spatial autocorrelation descriptors (**ATS centered)**
  - the index of spatial correlation aka « **Moran »**
  - The distance correlation function aka « **Geary »**

Broto,P., Moreau,G. and Vandicke,C., 1984, Eur. J. Med. Chem., 19, 71–78.

Some equations

Moreau Broto

$$ATS_d = \sum_{i=1}^{A}\sum_{j=1}^{A} \delta_{ij} \cdot (w_i \cdot w_j)_d = \mathbf{W}^T \cdot \mathbb{B}^m \cdot \mathbf{W}$$

Moreau Broto centered

$$\overline{ATS}_d = \frac{1}{\Delta} \cdot \sum_{i=1}^{A}\sum_{j=1}^{A} \delta_{ij} \cdot (w_i \cdot w_j)_d$$

dependence on the molecular size

Moran

$$I(d) = \frac{\frac{1}{\Delta} \cdot \sum_{i=1}^{A}\sum_{j=1}^{A} \delta_{ij} \cdot (w_i - \overline{w}) \cdot (w_j - \overline{w})}{\frac{1}{A} \cdot \sum_{i=1}^{A} (w_i - \overline{w})^2}$$

Geary

$$c(d) = \frac{\frac{1}{2\Delta} \cdot \sum_{i=1}^{A}\sum_{j=1}^{A} \delta_{ij} \cdot (w_i - w_j)^2}{\frac{1}{A-1} \cdot \sum_{i=1}^{A} (w_i - \overline{w})^2}$$

$[0, \infty]$

Example of Scripts

```
from __future__ import print_function
from rdkit import Chem
from rdkit.Chem import rdMolDescriptors as rdMD
from rdkit.Chem import AllChem
m = Chem.MolFromSmiles('Cc1ccccc1')
m2 = Chem.AddHs(m)
AllChem.EmbedMolecule(m2,AllChem.ETKDG())
m2 = Chem.RemoveHs(m2)
rdMD.CalcAUTOCORR2D(m)
```

```
>>> rdMD.CalcAUTOCORR2D(m)
[2.079, 2.197, 1.792, 0.693, 0.0, 0.0, 0.0, 0.0, 2.079, 2.197, 1.792,
 0.693, 0.0, 0.0, 0.0, 0.0, 2.079, 2.197, 1.792, 0.693, 0.0, 0.0, 0.0
, 0.0, 2.079, 2.197, 1.792, 0.693, 0.0, 0.0, 0.0, 0.0, 2.079, 2.197,
1.792, 0.693, 0.0, 0.0, 0.0, 0.0, 4.382, 4.458, 3.864, 1.609, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0
.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
 0.0, 8.232, 8.126, 5.768, 1.705, 0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 1.0,
1.0, 0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 1.0, 1.0, 0.0, 0.0, 0.0, 0.0, 1.0,
 1.0, 1.0, 1.0, 0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 1.0, 1.0, 0.0, 0.0, 0.0
, 0.0, 1.0, 1.0, 1.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.332, -0.294, -0.465,
 -1.15, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0
.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
 0.0, 0.0, 0.0, 0.0, 0.468, 1.044, 1.355, 2.601, 0.0, 0.0, 0.0, 0.0]
```

AUTOCORR2D Vector output definition (size: 192)

Four Methods:

ATS : **Moreau Broto**
ATSc : **Moreau Broto Centered**
MATS: **Moran**
GATS : **Geary**

For each Methods

Six groups (for 6 atomic properties):

Position	Code	Name
1-8	ATS/m*	Relative atomic Mass
9-16	ATS/v	Relative van der Waals volume
17-24	ATS/e	Relative Electronegativity
25-32	ATS/p	Relative atomic polarizability
33-40	ATS/i	Relative atomic ion polarity
41-48	ATS/s	Relative IState

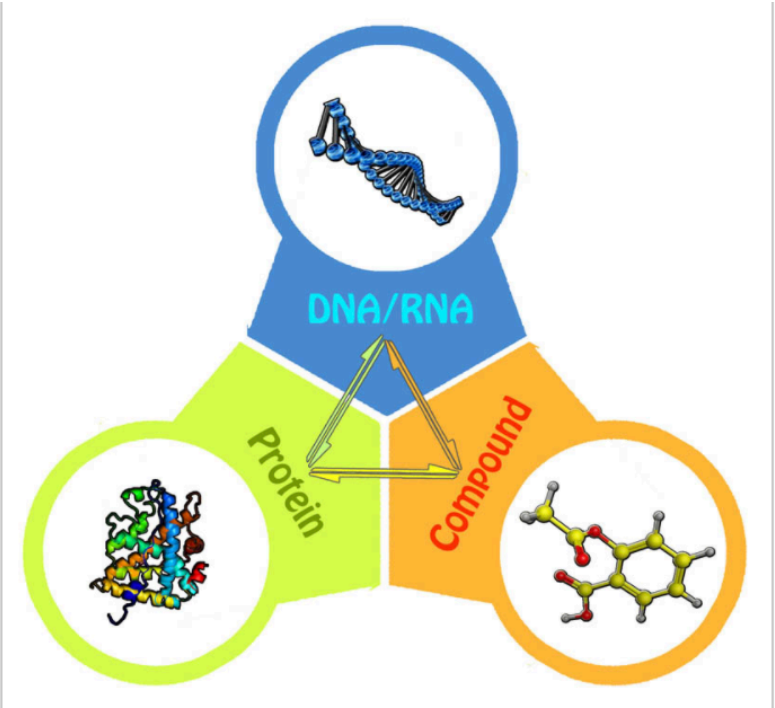
Eight AC for each groups

Some Applications of AUTOCORR2D:

- Prediction of protein helix content from an autocorrelation analysis of sequence hydrophobicities
- Accurate prediction of protein secondary structural content
- Prediction of membrane protein types based on the hydrophobic index of amino acids
- Population structure inferred by local spatial autocorrelation: an example from an Amerindian tribal population
- Prediction of transporter family from protein sequence by support vector machine approach
- Prediction of protein folding class using global description of amino acid sequence

Li,Lin,Han,Jiang,Chen,Chen ,Nucleic Acids Res. 2006 Jul 1; 34(Web Server issue): W32–W37. 10.1093/nar/gkl305

BioTriangle (http://biotriangle.scbdd.com/)



Tools	Features	Description
BioChem	• Constitution(30) • Connectivity descriptors (44) • Topology descriptors (35) • Kappa descriptors (7) • E-state descriptors (237) • Moran autocorrelation descriptors (32) • Geary autocorrelation descriptors (32) • Molecular property descriptors (6) • Moreau-Broto autocorrelation descriptors (32) • Charge descriptors (25) • MOE-type descriptors (60) • Daylight-type fingerprints • MACCS fingerprints • Atom pairs fingerprints • Morgan fingerprints • TopologicalTorsion fingerprints • E-state fingerprints • FP4 fingerprints	Details>>
BioProt	• Amino acid composition (20) • Dipeptide composition (400) • Tripeptide composition (8000) • CTD composition (21) • CTD transition (21) • CTD distribution (105) • M-B autocorrelation (240) • Moran autocorrelation (240) • Geary autocorrelation (240) • Conjoint triad features (343) • Quasi-sequence order descriptors (100) • Sequence order coupling number (60) • Pseudo amino acid composition 1 (50) • Pseudo amino acid composition 2 (50)	Details>>

RDKit GETAWAY:

- Defined from « GEometry, Topology, and Atom-Weights AssemblY »
- Majority of those descriptors are closely related to autocorrelation 3D descriptors computed using leverage matrix obtained by the centered atomic coordinates (molecular influence matrix MIM) and the (influence/distance matrix R).
- They encode information on structural fragments using spacial autocorrelation
- They encode specific atom properties shape & molecular size

V. Consonni, R. Todeschini, M. Pavan, J. Chem. Inf. Comput. Sci. 2002, 42, 682–692

GETAWAY Vector output definition (size: 273)

Parameters based on Matrix operations & information indices

Position	Code	Name	Method
1	ITH	Total information content on leverage equality	Based on cluster of similar leverage values
2	ISH	Standardized information content on leverage equality	Based on cluster of similar leverage values
3	HIC	Mean Information content on leverage equality	Use linear, planar or non-planar parameter D
4	HGM	Geometric mean on Leverage magnitude	4
145	RCON	R-connectivity index	Depend on the Bond number
146	RARS	Average row sunm of the influence/distance matrix R	
147	REIG	First eigenvalue of the R matrix	

Some Equations

Formula	Eq. no.	Name
$H_{GM} = 100 \cdot \left(\prod_{i=1}^{A} h_{ii} \right)^{1/A}$	(32)	Geometric mean on the leverage magnitude
$I_{TH} = A_0 \cdot \log_2 A_0 - \sum_{g=1}^{G} N_g \cdot \log_2 N_g$	(33)	Total information content on the leverage equality
$I_{SH} = \frac{I_{TH}}{A_0 \cdot \log_2 A_0} = 1 - \frac{\sum_{g=1}^{G} N_g \cdot \log_2 N_g}{A_0 \cdot \log_2 A_0}$	(34)	Standardized information content on the leverage equality
$HIC = \bar{I}_H = -\sum_{i=1}^{A} \frac{h_{ii}}{D} \cdot \log_2 \frac{h_{ii}}{D}$	(35)	Mean information content on the leverage magnitude
$RARS = \frac{1}{A} \cdot \sum_{i=1}^{A} \sum_{j=1}^{A} \frac{\sqrt{h_{ii} \cdot h_{jj}}}{r_{ij}} = \frac{1}{A} \cdot \sum_{i=1}^{A} RS_i$	(36)	Average row sum of the influence/distance matrix
$RCON = \sum_{b=1}^{B} (RS_i \cdot RS_j)_b^{1/2}$	(37)	**R**-connectivity index
$REIG = \lambda_1$	(38)	First eigenvalue of the **R** matrix

Some properties are based on Cluster analysis:

ITH (33) and ISH (34)

We are not able to exactly reproduce the Clustering algorithm implemented in Dragon due to lack of details. Thus they can be as is.

We included them in RDKit as a initial implentation stage.

GETAWAY Vector output definition (size: 273)

Parameters based on autocorelation functions

Position	Code	Name
u15-23,m35-43,v55-63,e75-83,p95-103,i115-123,s135-143	HATSk(w)*	HATS indices (similar to ATS)
u24,m44,v64,e84,p104,i124,s144	HATS(w)	HATS total index
u5-13,m25-33,v45-53,e65-73,p85-93,i105-113,s125-133	Hk(w)	H indices
u14,m34,v54,e74,p94,i114,s134	HT(w)	H total index
u148-155,m166-173,v184-191,e202-209,p220-227,i238-245,s256-263	Rk(w)	R indices
u156,m174,v192,e210,p228,i246,s264	RT(w)	R total index
u157-164,m175-182,v193-200,e211-218,p229-236,i247-254,s256-272	R+k(w)	Maximal R indices
u165,m183,v201,e219,p237,i255,s273	RT+(w)	Maximal R total index

Unweighted, Relative atomic Mass, Relative van der Waals volume, Relative Electronegativity, Relative atomic polarizability
Relative atomic ion polarity , Relative IState

Some equations

Formula	*Eq. no.*	*Name*
$HATS_k(w) = \sum_{i=1}^{A-1} \sum_{j>i} (w_i \cdot h_{ii}) \cdot (w_j \cdot h_{jj}) \cdot \delta(k; d_{ij}) \quad k = 0, 1, 2, \ldots, d$	(39)	HATS indices
$HATS(w) = HATS_0(w) + 2 \cdot \sum_{k=1}^{d} HATS_k(w)$	(40)	HATS total index
$H_k(w) = \sum_{i=1}^{A-1} \sum_{j>i} h_{ij} \cdot w_i \cdot w_j \cdot \delta(k; d_{ij}; h_{ij}) \quad k = 0, 1, 2, \ldots, d$	(41)	**H** indices
$HT(w) = H_0(w) + 2 \cdot \sum_{k=1}^{d} H_k(w)$	(42)	**H** total index
$R_k(w) = \sum_{i=1}^{A-1} \sum_{j>i} \frac{\sqrt{h_{ii} \cdot h_{jj}}}{r_{ij}} \cdot w_i \cdot w_j \cdot \delta(k; d_{ij}) \quad k = 1, 2, \ldots, d$	(43)	**R** indices
$RT(w) = 2 \cdot \sum_{k=1}^{d} R_k(w)$	(44)	**R** total index
$R_k^+(w) = \max_{ij} \left(\frac{\sqrt{h_{ii} \cdot h_{jj}}}{r_{ij}} \cdot w_i \cdot w_j \cdot \delta(k; d_{ij}) \right) \quad i \neq j \text{ and } k = 1, 2, \ldots, d$	(45)	Maximal **R** indices
$RT^+(w) = \max_k(R_k^+(w))$	(46)	Maximal **R** total index

http://michem.disat.unimib.it/chm/download/materiale/geometrical_descriptors.pdf Todeschini, Consonni

Example of Scripts

```
from __future__ import print_function
from rdkit import Chem
from rdkit.Chem import rdMolDescriptors as rdMD
from rdkit.Chem import AllChem
m = Chem.MolFromSmiles('Cc1ccccc1')
m2 = Chem.AddHs(m)
AllChem.EmbedMolecule(m2,AllChem.ETKDG())
m2 = Chem.RemoveHs(m2)
rdMD.CalcGETAWAY(m2)
```

```
>>> rdMD.CalcGETAWAY(m2)
[19.651, 1.0, 3.385, 13.18, 3.0, 1.212, 1.0, 0.696, 0.258, 0.0, 0.0,
0.06, -2.077, 5.3, 1.136, 0.249, 0.814, 0.537, 1.082, 0.964, 0.285, 0
.0, 0.0, 9.0, 0.502, 0.335, 0.071, 0.016, 0.002, 0.0, 0.0, -0.076, -0
.167, 0.864, 0.047, 0.046, 0.051, 0.057, 0.045, 0.018, 0.002, 0.0, 0.
0, 0.484, 0.659, 0.506, 0.228, 0.077, 0.018, 0.0, 0.0, -0.841, 2.035,
 4.706, 0.115, 0.086, 0.118, 0.126, 0.156, 0.094, 0.02, 0.0, 0.0, 1.3
16, 2.716, 1.156, 0.932, 0.626, 0.229, 0.0, 0.0, -0.109, -2.051, 4.28
1, 1.012, 0.236, 0.735, 0.497, 0.982, 0.863, 0.252, 0.0, 0.0, 8.143,
0.849, 0.619, 0.338, 0.136, 0.037, 0.0, 0.0, 0.0, 0.0, 3.108, 0.198,
0.112, 0.184, 0.177, 0.253, 0.173, 0.041, 0.0, 0.0, 2.08, 4.153, 1.41
1, 1.254, 0.978, 0.377, 0.0, 0.0, 0.046, 0.048, 12.382, 1.639, 0.295,
 1.132, 0.691, 1.48, 1.371, 0.415, 0.0, 0.0, 12.406, 4.441, 2.91, 1.7
76, 0.844, 0.258, 0.0, 0.0, 0.02, 0.003, 16.064, 1.254, 0.552, 1.065,
 0.94, 1.484, 1.106, 0.285, 0.713, 0.076, 13.697, 10.817, 0.834, 0.96
5, 1.427, 1.558, 1.194, 1.285, 0.636, 0.153, 0.0, 0.0, 12.506, 0.24,
0.349, 0.089, 0.131, 0.089, 0.066, 0.0, 0.0, 0.349, 0.382, 0.26, 0.17
6, 0.068, 0.016, 0.001, 0.0, 0.0, 1.807, 0.064, 0.039, 0.026, 0.021,
0.004, 0.0, 0.0, 0.0, 0.064, 0.587, 0.418, 0.329, 0.205, 0.073, 0.011
, 0.0, 0.0, 3.244, 0.064, 0.039, 0.026, 0.021, 0.014, 0.005, 0.0, 0.0
, 0.064, 1.361, 1.439, 1.112, 1.169, 0.573, 0.136, 0.0, 0.0, 11.577,
0.226, 0.309, 0.079, 0.116, 0.079, 0.059, -0.005, -0.156, 0.309, 0.72
1, 0.547, 0.441, 0.321, 0.127, 0.022, 0.0, 0.0, 4.358, 0.091, 0.051,
0.028, 0.024, 0.02, 0.01, 0.0, 0.0, 0.091, 1.664, 2.022, 1.506, 1.74,
 0.891, 0.223, 0.0, 0.0, 16.091, 0.29, 0.509, 0.13, 0.191, 0.13, 0.09
7, 0.0, 0.0, 0.509, 3.39, 2.729, 2.244, 1.84, 0.785, 0.153, 0.0, 0.0,
 22.282, 0.48, 0.349, 0.146, 0.131, 0.107, 0.066, 0.0, 0.0, 0.48]
```

Some Applications of GETAWAY:

- HATS, H ,R and maximal R indices are molecular descriptors for structural property correlations
- Also used for molecular profiles suitable for similarity/diversity analysis studies

http://michem.disat.unimib.it/chm/download/materiale/geometrical_descriptors.pdf Todeschini, Consonni

RDKit WHIM:

- Defined by « Weighted Holistic Invariant Molecular » descriptors are geometrical descriptors based on statistical indices calculated on the projections of the atoms along principal axes (using SVD analysis).
- There are two type directional and global
- Algorithm consists in calculating the eigenvalues and eigenvectors of a weighted covariance matrix of the centered Cartesian coordinates of a molecule, obtained from different weighting schemes for the atoms (unweighted, atomic mass, the van der Waals volume, Sanderson atomic electronegativity, atomic polarizability, electrotopological state indices S of Kier and Hall)

Some Equations

Formula	*Eq. no.*	*Name*	*Molecular feature*
$\lambda_m \quad m = 1, 2, 3$	(19)	d-WSIZ indices	Axial dimension
$\mathrm{T} = \lambda_1 + \lambda_2 + \lambda_3$	(20)	WSIZ index	Global dimension
$\mathrm{A} = \lambda_1\lambda_2 + \lambda_1\lambda_3 + \lambda_2\lambda_3$	(21)	WSIZ index	Global dimension
$\mathrm{V} = \prod_{m=1}^{3}(1 + \lambda_m) - 1 = \mathrm{T} + \mathrm{A} + \lambda_1\lambda_2\lambda_3$	(22)	WSIZ index	Global dimension
$\vartheta_m = \frac{\lambda_m}{\sum_m \lambda_m} \quad m = 1, 2, 3$	(23)	d-WSHA indices	Axial shape
$K = \frac{3}{4} \cdot \sum_{m=1}^{3} \left\lvert \frac{\lambda_m}{\sum_m \lambda_m} - \frac{1}{3} \right\rvert$	(24)	WSHA index	Shape
$\eta_m = \frac{\lambda^2 \cdot A}{\sum_i t_i^4} \quad m = 1, 2, 3$	(25)	d-WDEN indices	Axial density
$\mathrm{D} = \eta_1 + \eta_2 + \eta_3$	(26)	WDEN index	Global density
$\gamma_m = \left\{1 - \left[\frac{n_s}{A} \cdot \log_2 \frac{n_s}{A} + n_a \cdot \left(\frac{1}{A} \cdot \log_2 \frac{1}{A}\right)\right]\right\}^{-1} \quad m = 1, 2, 3$	(27)	d-WSYM indices	Axial symmetry
$\mathrm{G} = (\gamma_1 \cdot \gamma_2 \cdot \gamma_3)^{1/3}$	(28)	WSYM index	Symmetry

Some properties are based on Cluster analysis:

Gamma's (27) and Gamma (28)

We are not able to exactly reproduce the Clustering algorithm implemented in Dragon due to lack of details. Thus they can be as is.

We included them in RDKit as a initial implentation stage.

http://michem.disat.unimib.it/chm/download/materiale/geometrical_descriptors.pdf Todeschini, Consonni

WHIM Vector output definition (size: 114)

Directionals Descriptors

Position	Code	Name
1-11	L1u,L2u,L3u,P1u,P2u,G1u,G2u,G3u,E1u,E2u,E3u	unweighted
12-22	L1m,L2m,L3m,P1m,P2m,G1m,G2m,G3m,E1m,E2m,E3m	Relative atomic Mass
23-33	L1v,L2v,L3v,P1v,P2v,G1v,G2v,G3v,E1v,E2v,E3v	Relative van der Waals volume
34-44	L1e,L2e,L3e,P1e,P2e,G1e,G2e,G3e,E1e,E2e,E3e	Relative Electronegativity
45-55	L1p,L2p,L3p,P1p,P2p,G1p,G2p,G3p,E1p,E2p,E3p	Relative atomic polarizability
56-66	L1i,L2i,L3i,P1i,P2i,G1i,G2i,G3i,E1i,E2i,E3i	Relative atomic ion polarity
67-77	L1s,L2s,L3s,P1s,P2s,G1s,G2s,G3s,E1s,E2s,E3s	Relative IState

L= lambda : axial dimension, P= : axial shape, G = gamma : axial symetry, E = nu : axial density

We do not add by definition: P3 = 1-P1+P2

G1, G2, G3 are not always identical to Dragon values

WHIM Vector output definition (size:114)

Global Descriptors

Position	Code	Name
78-84	Tu,Tm,Tv,Te,Tp,Ti,Ts	Lamdba sum (20)
85-91	Au,Am,Av,Ae,Ap,Ai,As	Sum Cross Lamdba products (21)
92-93	Gu,Gm	Sum Gamma (28)
94-100	Ku,Km,Kv,Ke,Kp,Ki,Ks	Inverse Kurtosis (24)
101-108	Du,Dm,Dv,De,Dp,Di,Ds	Sum Density (25)
108-114	Vu,Vm,Vv,Ve,Vp,Vi,Vs	Sum of Lambda all products (22)

For G only unweighted and relative mass parameters are include in Dragon Version

G is not always identical to Dragon

Example of Scripts

```
from __future__ import print_function
from rdkit import Chem
from rdkit.Chem import rdMolDescriptors as rdMD
from rdkit.Chem import AllChem
m = Chem.MolFromSmiles('Cc1ccccc1')
m2 = Chem.AddHs(m)
AllChem.EmbedMolecule(m2,AllChem.ETKDG())
m2 = Chem.RemoveHs(m2)
rdMD.CalcWHIM(m2,thresh=0.01)
```

Default threshold is 0.001

```
>>> rdMD.CalcWHIM(m2)
[3.421, 1.698, 0.103, 0.655, 0.325, 0.204, 0.204, 0.258, 0.568, 0.471
, 0.129, 2.175, 0.95, 0.017, 0.692, 0.302, 0.204, 0.204, 0.204, 0.23,
 0.147, 0.003, 2.577, 1.191, 0.045, 0.676, 0.312, 0.204, 0.204, 0.242
, 0.322, 0.232, 0.024, 3.379, 1.673, 0.1, 0.656, 0.325, 0.204, 0.204,
 0.278, 0.554, 0.457, 0.121, 2.777, 1.312, 0.059, 0.67, 0.316, 0.204,
 0.204, 0.242, 0.375, 0.281, 0.041, 3.551, 1.776, 0.112, 0.653, 0.327
, 0.204, 0.204, 0.258, 0.612, 0.515, 0.152, 2.984, 1.435, 0.072, 0.66
4, 0.32, 0.204, 0.204, 0.229, 0.432, 0.336, 0.061, 5.222, 3.142, 3.81
3, 5.153, 4.148, 5.44, 4.49, 6.338, 2.119, 3.238, 6.161, 3.884, 6.906
, 4.598, 0.221, 0.204, 0.483, 0.538, 0.514, 0.484, 0.504, 0.479, 0.49
7, 0.389, 0.127, 0.193, 0.378, 0.232, 0.427, 0.277, 12.161, 5.297, 7.
189, 11.882, 8.246, 13.054, 9.395]
```

```
>>> rdMD.CalcWHIM(m2,thresh=0.01)
[3.421, 1.698, 0.103, 0.655, 0.325, 0.209, 0.218, 0.422, 0.568, 0.471
, 0.129, 2.175, 0.95, 0.017, 0.692, 0.302, 0.209, 0.218, 0.258, 0.23,
 0.147, 0.003, 2.577, 1.191, 0.045, 0.676, 0.312, 0.209, 0.218, 0.371
, 0.322, 0.232, 0.024, 3.379, 1.673, 0.1, 0.656, 0.325, 0.209, 0.218,
 0.422, 0.554, 0.457, 0.121, 2.777, 1.312, 0.059, 0.67, 0.316, 0.209,
 0.242, 0.422, 0.375, 0.281, 0.041, 3.551, 1.776, 0.112, 0.653, 0.327
, 0.209, 0.218, 0.422, 0.612, 0.515, 0.152, 2.984, 1.435, 0.072, 0.66
4, 0.32, 0.209, 0.242, 0.422, 0.432, 0.336, 0.061, 5.222, 3.142, 3.81
3, 5.153, 4.148, 5.44, 4.49, 6.338, 2.119, 3.238, 6.161, 3.884, 6.906
, 4.598, 0.268, 0.228, 0.483, 0.538, 0.514, 0.484, 0.504, 0.479, 0.49
7, 0.389, 0.127, 0.193, 0.378, 0.232, 0.427, 0.277, 12.161, 5.297, 7.
189, 11.882, 8.246, 13.054, 9.395]
```

Some Applications of WHIM:

- WHIM descriptors have been used to model toxicological indices
- Several physicochemical properties of PCBs and PAHs
- Hydroxyl radical reaction rate constants
- Soil sorption partition coefficients

http://michem.disat.unimib.it/chm/download/materiale/geometrical_descriptors.pdf Todeschini, Consonni

Conclusion

Name	Number	Dragon v6 comparison
RDF	210	identical
MORSE	224	identical
AUTOCORR3D	80	identical
AUTOCORR2D	192	identical
GETAWAY	273	Possible variation on cluster data
WHIM	114	Possible variation on cluster data
Total	**1093**	

Adding Customs Atoms Properties

If CustomAtomProperty doesn't set than custom weigth are set to 1 (ie unweigthed)

```
>>> rdMD.CalcRDF(m2,
CustomAtomProperty="AAA")
[3.143, 2.462, 2.172, 9.224, 0.569,
3.7, 2.12, 2.964, 1.722, 1.618,
0.384, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
```

Name	Length of Custom Vector
RDF	30
MORSE	32
AUTOCORR3D	10
AUTOCORR2D	32
GETAWAY	45
WHIM	17

```
atoms = m2.GetNumAtoms()
for idx in range( atoms ):
        m2.GetAtomWithIdx( idx ).SetDoubleProp( 'molAtomMass', m2.GetAtomWithIdx( idx ).GetMass()  )
```

```
rdMD.CalcAUTOCORR3D(m2,
CustomAtomProperty="molAtomMass")
[14.433, 29.854, 28.472, 12.531, 2.686,
0.171, 0.0, 0.0, 0.0, 0.0]
```

```
 rdMD.CalcRDF(m2,
CustomAtomProperty="molAtomMass")
[37.073, 346.92, 24.115, 584.871, 24.357,
34.465, 30.643, 30.883, 3.58, 6.327,
1.025, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0]
```

EEM partial Charges in RDKit

- Partial Charges EEM is a good example of the Predicition of QM result using modeling:
  - Electronegativity equalization « empirical » method mimic QM atomic partial charges

$$\begin{bmatrix} B_1 & \frac{\kappa}{R_{1,2}} & \cdots & \frac{\kappa}{R_{1,N}} & -1 \\ \frac{\kappa}{R_{2,1}} & B_2 & \cdots & \frac{\kappa}{R_{2,N}} & -1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \frac{\kappa}{R_{N,1}} & \frac{\kappa}{R_{N,2}} & \cdots & B_N & -1 \\ 1 & 1 & \cdots & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} q_1 \\ q_2 \\ \vdots \\ q_N \\ \bar{\chi} \end{bmatrix} = \begin{bmatrix} -A_1 \\ -A_2 \\ \vdots \\ -A_N \\ Q \end{bmatrix}$$

Racek, Pazurikova, Varekova, Geidl, Krenek, Falginella, Horsky, Hejret and Koca *J. of Cheminfo.* 2016, 8:57 https://jcheminf.springeropen.com/articles/10.1186/s13321-016-0171-1

EEM in RDKit

```
from __future__ import print_function
from rdkit import Chem
from rdkit.Chem import rdMolDescriptors as rdMD
from rdkit.Chem import AllChem
m = Chem.MolFromSmiles('Cc1ccccc1')
m2 = Chem.AddHs(m)
AllChem.EmbedMolecule(m2,AllChem.ETKDG())
rdMD.CalcEEMcharges(m2)
```

```
>>> rdMD.CalcEEMcharges(m2)
[-0.47045274173257656, 0.09949541803432527
, -0.3064804676209105, -0.1633383931726717
4, -0.24288863496349222, -0.15346886725900
7, -0.3238900196021737, 0.1907501122162559
3, 0.1941896477791849, 0.19155802321573207
, 0.20196220411193408, 0.18822579267908232
, 0.19828449311845903, 0.18849995007175582
, 0.20755348312410327]
```