Table of Contents

Introduction

This document will be divided into the following sub-sections:

- **Libraries and Tools:**
  In this section I will describe a lot of tools that could be used for generating chemical descriptors for different chemical compounds. At first I will discuss each tool/library in details to illustrate why I think it's useful for the EPA ToxCast project, refer to the input/output formats, licenses, developer guide, programming language used, how to get it some examples if possible, and. At the end of this section I will add a summary table which contains all of these tools in a summary manner.
- **Input/output File Formats:**
  In this section I will present a table which contains different input and output formats with a sample file shows the format for each type.
- **Ideas and Approaches:**
  In this section I will provide some approaches, ideas and research work that I believe **(from my prospective as a non-expert in this field)** that it may be useful for the EPA ToxCast project.
- **Databases:**
  This section will contain a list of databases for different chemical compounds descriptors.
- **Helper Tools:**
  This section will contain a list of tools that may be helpful for future competitions.
- **References:**
  In this section I will list some references and materials that helped me to get started and understand more about cheminformatics in short time. I think it could be useful to help competitors in future contests to get started.

Libraries and Tools

The Chemistry Development Kit (CDK)

Summary

The Chemistry Development Kit (CDK) is a Java library for structural chemo- and bioinformatics. It is now developed by more than 50 developers all over the world and used in more than 10 different academic as well as industrial projects worldwide.

Why this may be useful

This library provides the following services:

- **Ability to perform QSAR descriptor calculation on chemical compounds.**
- **Ability to perform substructure search using exact structures and SMARTS-like queries.**
- 2D diagrams editing and generation.
- 3D geometry generation.
- Structure generators.
- Fingerprint calculations.
- **Supports many chemical input/output formats, including SMILES, CML, and MDL formats.**
- **Ability to convert between different formats.**
- **International Chemical Identifier support.**

So I think this library maybe help us in different project phases if we used it as a base for our descriptor generation project as it contains a lot of helpful tools besides supporting the main functionality required in the project.

Programming Language

This library provides APIs in Java programming language however there are some wrappers interfaces for other programming languages like (Python, Ruby) will be discussed later.

Example

The CDK has been extended with the addition of the cdk.qsar module that allows for the calculation of molecular descriptors. Currently 33 descriptors are present covering **topolgical, geometric and electronic descriptor** classes. The snippet below shows how you can obtain the value of a specific descriptor for a single molecule.

```
IAtomContainer ac;

IDescriptor descriptor = new WienerNumbersDescriptor();

DoubleArrayResult retval = (DoubleArrayResult)descriptor.calculate(ac);

double wpath = retval.get(0); // Wiener path number

double wpol = retval.get(1);  // Wiener polarity number
```

In addition to the calculation of individual descriptors it is also possible to evaluate all descriptors or subsets of descriptors implemented in the CDK. This is performed by the DescriptorEngine. To calculate all available descriptors we can use the code below

```
Molecule molecule;

// initialize the Molecule object

DescriptorEngine engine = new DescriptorEngine();

engine.process(molecule);
```

In case we want to calculate specific classes of descriptors (say topological and geometric) we can do

```
String[] types = {'topological','geometric'};

DescriptorEngine engine = new DescriptorEngine(types);

engine.process(molecule);
```

In both cases, the result of each descriptor is stored in the Molecule object as a DescriptorValue object keyed on the DescriptorSpecification object for that descriptor.

Input/output

A common task is to read a molecule from a disk file. Since the CDK supports a number of file formats, one can read in a specific format or use more general code to automatically detect the format. In the first case the code to read in the molecules in an SD file would be

```
String filename = "molecules.sdf";
InputStream ins = this.getClass().getClassLoader().getResourceAsStream(filename);
MDLReader reader = new MDLReader(ins);
// alternatively, you can specify a file directly
// MDLReader sdfreader = new MDLV2000Reader(new FileReader(new File(filename)));
ChemFile chemFile = (ChemFile)reader.read((ChemObject)new ChemFile());
List containersList = ChemFileManipulator.getAllAtomContainers(chemFile);
```

However a more general code snippet below allows you to simply specify the filename and automatically detect the format and load the molecules

```
public static IAtomContainer[] loadMolecules(String[] filenames) throws CDKException {
    Vector v = new Vector();
    DefaultChemObjectBuilder builder = DefaultChemObjectBuilder.getInstance();
    try {
        int i;
        int j;
        for (i = 0; i < filenames.length; i++) {
            File input = new File(filenames[i]);
            ReaderFactory readerFactory = new ReaderFactory();
            IChemObjectReader reader = readerFactory.createReader(new FileReader(input));
            IChemFile content = (IChemFile) reader.read(builder.newChemFile());
            if (content == null) continue;
            List c = ChemFileManipulator.getAllAtomContainers(content);
            // we should do this loop in case we have files
            // that contain multiple molecules
            for (j = 0; j < c.size(); j++) v.add((IAtomContainer) c.get(j));
        }
    } catch (Exception e) {
        e.printStackTrace();
```

```
            throw new CDKException(e.toString());
        }
        // convert the vector to a simple array
        IAtomContainer[] retValues = new IAtomContainer[v.size()];
        for (int i = 0; i < v.size(); i++) {
            retValues[i] = v.get(i);
        }
        return retValues;
    }
```

Please refer to all possible input/output file formats in the (Input/Output File formats)

The output of this library could be descriptors calculations, similarity values or molecule formats.

To find out the available formats look under the cdk.io package.

License

This library is an open source tool under GNU Library or Lesser General Public License version 2.0 (LGPLv2).

How to get it

The download link could be found here.

Documentation

All APIs documentation could be found here.

Citations

The CDK has a lot of citations. You can find them all here.

Open Babel

Summary

Open Babel is a chemical toolbox designed to speak the many languages of chemical data. It's an open, collaborative project allowing anyone to search, convert, analyze, or store data from molecular modeling, chemistry, solid-state materials, biochemistry, or related areas.

Open Babel includes two components, a command-line utility and a C++ library. The command-line utility is intended to be used as a replacement for the original babel program, to translate between various chemical file formats. The C++ library includes all of the file-translation code as well as a wide variety of utilities to foster development of other open source scientific software.

In addition to serving as a set of user-level tools, Open Babel offers a C++ library and interface in other languages (e.g., Perl and Python for general chemical software development, both in-house and to encourage open source chemistry packages. As such, it is also a founding member of the Blue Obelisk movement, which shares cheminformatics data, algorithms and more).

Why this may be useful

This library provides the following services:

- **Read, write and convert over 110 chemical file formats.**
- **Filter and search molecular files using SMARTS and other methods.**
- **Supports molecular modeling, cheminformatics, bioinformatics.**
- **Implementation of Daylight SMARTS molecular matching syntax.**
- **Batch conversion for multiple molecules in one file (e.g., splitting, merging, batch operation).**
- Cross platform.

This library sounds reliable as it had been used as base for a lot of projects (check citation section below), also it contains a lot of utilities that can facilitate a lot of work for system developers.

Programming Language

Open Babel offers a C++ library and interface in other languages (e.g., Perl and Python for general chemical software development).

Example

Examples on using the command line tool:

File Conversion:

- To convert mymols.sdf to SMILES format.

PROMPT> babel -isdf 'mymols.sdf' -osmi 'outputfile.smi'

- Multiple input files can be converted in batch format too. To convert all files ending in .xyz (*.xyz) to PDB files, you can type:

PROMPT> babel *.xyz -opdb –m

- If you only want to convert a subset of molecules you can define them using -f and -l, so to convert molecules 2-4 of the file mymols.sdf type:

PROMPT> /babel 'mymols.sdf' -f 2 -l 4 -osdf 'outputfile.sdf'

- Alternatively you can select a subset matching a SMARTS pattern, so to select all molecules containing bromobenzene use:

PROMPT> babel mymols.sdf -osdf 'selected.sdf' -s 'c1ccccc1Br'

Molecular fingerprints:

- You can see the available fingerprints by typing the following command:

PROMPT> babel -L fingerprints
FP2 Indexes linear fragments up to 7 atoms.
FP3 SMARTS patterns specified in the file patterns.txt
FP4 SMARTS patterns specified in the file SMARTS_InteLigand.txt
MACCS SMARTS patterns specified in the file MACCS.txt

Similarity searching:

For relatively small datasets (<10,000's) it is possible to do similarity searches without the need to build a similarity index, however larger datasets (up to 100,000's) can be searched rapidly once a fastsearch index has been built.
On larger datasets it is necessary to first build a fastsearch index. This is an new file that stores a database of fingerprints for the files indexed. You will still need to keep both the new .fs fastsearch index and the original files. However, the new index will allow significantly faster searching and similarity comparisons. The index is created with the following command:

```
PROMPT>  babel mymols.sdf –ofs
```

This builds mymols.fs with the default fingerprint (unfolded). The following command uses the index to find the 5 most similar molecules to the molecule in query.mol:

```
PROMPT>  babel mymols.fs results.sdf -S query.mol -at 5
```

For more examples check this.

Input/output

The Open Babel library and tools support import and export for a wide range of chemical file formats and data types. The following link contains an index to all formats, organized by the name of the format.

License

This library is an open source tool under GNU Library or Lesser General Public License version 2.0 (LGPLv2).

How to get it

The download link could be found here.

Documentation

All APIs documentation could be found here.

Installation

For installation steps for both (Windows and Linux) check this page.

Citations

This library downloaded over **164,000 times** and used by over **40 related projects.** A list of citations for this library could be found here.

Opentox

Summary

OpenTox is an Open Source platform for predictive toxicology. The goal of OpenTox is to develop an interoperable predictive toxicology framework which may be used as an enabling platform for the creation of predictive toxicology applications.

Why this may be useful

OpenTox is a framework for the integration of algorithms for predicting chemical toxicity using the following approaches:

- Components for specialized tasks (e.g. database lookups, descriptor calculation, classification, regression, report generation) that communicate through well-defined language independent interfaces.

- The framework supports building multiple applications, as well as providing components for third party applications.
- The framework guarantees the portability of components by enforcing language independent interfaces. Implementation of an integration component in a specific language/platform automatically ports the entire OpenTox framework to that language/platform.

This framework touches exactly the area of EPA ToxCost project. It's designated for the purpose of building predictive toxicology applications. Also it provides a standard REST interfaces for any programming language.

You can find a list of components this framework contains.

Programming Language

Web-services with a REST interface.

Example

Examples for the HTTP requests content

Request a compound in SDF format:

```
curl -X GET -H "Accept:chemical/x-mdl-sdfile" http://{server}/compound/{id}
```

Submit a compound in InChI format:

```
curl -X POST -H "Content-Type:chemical/x-inchi" --data-binary "InChI=1S/C5H10/c1-2-4-5-3-1/h1-5H2"
http://{server}/compound
```

File uploads:

Files can be uploaded by specifying "**multipart/form-data**" in the **Content-Type header**.

For more examples please check the APIs documentation page.

Input/output

Input will be a GET, POST, PUT or DELETE HTTP request to the REST APIs.

Output will be in the HTTP response, it could be a URL for a file contains the calculation result or a representation of a query result in a supported MIME type.

How to get it

All download links can be found here.

Documentation

APIs documentation can be found here.

OpenTox Tutorials.

License

Eclipse Public License (EPL)

ChemmineR

Summary

ChemmineR is a cheminformatics package for analyzing drug-like small molecule data in R. Its latest version contains functions for efficient processing of large numbers of molecules, physicochemical/structural property predictions, structural similarity searching, classification and clustering of compound libraries with a wide spectrum of algorithms. In addition, it offers visualization functions for compound clustering results and chemical structures.

The integration of chemoinformatic tools with the R programming environment has many advantages, such as easy access to a wide spectrum of statistical methods, machine learning algorithms and graphic utilities.

Why this may be useful

This package provides the following features:

- **Functions for efficient processing of large numbers of molecules.**
- **Physicochemical/structural property predictions.**
- **Structural similarity searching.**
- **Classification and clustering of compound libraries.**
- Visualization functions for compound clustering results and chemical structures.
- Various utilities for managing complex compound data.

This package contains a lot of useful clustering functions. I believe clustering is a main required functionality in the project to make more accurate predictions based on similarity between structure of compounds thus allowing to predict the response of an unknown chemical whose structure is much similar to a compound for which all information is available.

Programming Language

This package is built on R programming language.

Example

SDF Import:

```
## Import SD file and store as SDFset object
## Sample data set contains first 100 compounds from PubChem SD file
"Compound_00650001_00675000.sdf.gz"
## from: ftp://ftp.ncbi.nih.gov/pubchem/Compound/CURRENT-Full/SDF/
sdfset <-
read.SDFset("http://faculty.ucr.edu/~tgirke/Documents/R_BioCond/Samples/sdfsample.sdf")
data(sdfsample); sdfset <- sdfsample # The sample SDF set is provided by the library
valid <- validSDF(sdfset); which(!valid) # Identifies invalid SDFs in SDFset objects.
sdfset <- sdfset[valid] # Removes invalid SDFs, if there are any.

## Import SD file and store as SDFstr object
sdfstr <-
read.SDFstr("http://faculty.ucr.edu/~tgirke/Documents/R_BioCond/Samples/sdfsample.sdf")

## Create SDFset from SDFstr class
```

```
read.SDFset(sdfstr)
as(sdfstr, "SDFset")
```

SMILES Import:

```
## Create sample SMILES file for import
data(smisample); smiset <- smisample
write.SMI(smiset[1:4], file="sub.smi")

## Import SMILES file
smiset <- read.SMIset("sub.smi")

## Inspect content
smiset
An instance of "SMIset" with 100 molecules

view(smiset[1:2])
$`650001`
An instance of "SMI"
[1] "O=C(NC1CCCC1)CN(c1cc2OCCOc2cc1)C(=O)CCC(=O)Nc1noc(c1)C"

$`650002`
An instance of "SMI"
[1] "O=c1[nH]c(=O)n(c2nc(n(CCCc3ccccc3)c12)NCCCO)C"

## Accessor functions
cid(smiset[1:4])
[1] "650001" "650002" "650003" "650004"

(smi <- as.character(smiset[1:2]))
                                                    650001
"O=C(NC1CCCC1)CN(c1cc2OCCOc2cc1)C(=O)CCC(=O)Nc1noc(c1)C"
                                                    650002
         "O=c1[nH]c(=O)n(c2nc(n(CCCc3ccccc3)c12)NCCCO)C"

## Construct SMIset object from named vector
as(smi, "SMIset")
An instance of "SMIset" with 2 molecules
```

For more examples on descriptors functions and clustering examples check this page.

Input/output

SDF file format is extensively used in this package. The input will be molecules databases in SDF file formats and output will be results of performing any of its functions like clustering result, descriptors, similarity measures...etc.

License

How to get it

The R software for running ChemmineR can be downloaded here. The ChemmineR package itself is available at the Bioconductor repository. Due to its heavy development, it is strongly recommended to maintain a recent version of ChemmineR by updating or reinstalling it frequently.

Documentation

Documentation for this package could be found here and here.

Installation

For installation steps for (Windows, Linux and Mac) check this page.

RDKit

Summary

A collection of cheminformatics and machine-learning software written in C++ and Python.

The core algorithms and data structures are written in C++. Wrappers are provided to use the toolkit from either Python or Java.

Additionally, the RDKit distribution includes a PostgreSQL-based cartridge that allows molecules to be stored in relational database and retrieved via substructure and similarity searches.

Why this may be useful

This library provides the following services:

- Converting between SMILES or SDF and RDKit molecules
- Generating canonical SMILES
- Generating descriptors.
- Substructure filtering using SMARTS or RDKit molecules
- Substructure counter with visualization of counted substructures
- Highlighting atoms in molecules for, for example, showing the results of substructure matching
- Filtering sets of molecules by presence or absence of well-defined functional groups.
- Chemical reaction enumeration
- Stripping off salts from molecules
- Picking diverse molecule subsets
- R-group decomposition
- Generating Murcko scaffolds and frameworks
- Generating 2D coordinates for molecules, optionally including a template.
- Generating 3D coordinates for molecules.
- Generating a variety of molecular fingerprints and reading and writing fingerprint files:
  - RDKit fingerprints (Daylight-like topological fingerprint)
  - Morgan fingerprints (ECFP/FCFP-like circular fingerprints)
  - Atom pairs
  - Topological torsions
  - Avalon fingerprints

Programming Language

C++ and python

Example

Python Examples

Reading single molecules:
The majority of the basic molecular functionality is found in module rdkit.Chem

```
>>> from rdkit import Chem
```

Individual molecules can be constructed using a variety of approaches:

```
>>> m = Chem.MolFromSmiles('Cc1ccccc1')
>>> m = Chem.MolFromMolFile('data/input.mol')
>>> stringWithMolData=file('data/input.mol','r').read()
>>> m = Chem.MolFromMolBlock(stringWithMolData)
```

All of these functions return a Mol object on success:

```
>>> m
<rdkit.Chem.rdchem.Mol object at 0x...>
```

Reading sets of molecules
Groups of molecules are read using a Supplier (for example, an SDMolSupplier or a SmilesMolSupplier):

```
>>> suppl = Chem.SDMolSupplier('data/5ht3ligs.sdf')
>>> for mol in suppl:
...   print mol.GetNumAtoms()
...
20
24
24
26
```

For more python examples you can check this page.

Input/output

The input will be (2D or 3D) molecules files in SDF file format and output will be results of performing any of its functions like descriptors generation, similarity maps, descriptors visualization, chemical reactions...etc.

Also it can read a set of molecules in a compressed file.

License

BSD License

How to get it

Through the download link in this page.

Documentation

Python APIs documentation: here

C++ APIs documentation: here

Installation

Installation steps for (Windows, Linux and Mac) can be found here.

Indigo

Summary

Indigo is a universal organic chemistry toolkit. It contains first-class tools for end users, as well as a documented API for developers. Indigo it is completely free and open-source, while also available on a commercial basis.

Indigo is based on a cheminformatics library that incorporates a number of unique algorithms developed by GGA, as well as some standard algorithms well-known in the cheminformatics world. Since the core part of Indigo is written in modern C++ with no third-party code or dependencies except the ubiquitous zlib and libcairo, the toolkit provides outstanding performance and excellent portability.

Why this may be useful

This library provides the following services:

- Input formats support: Molfiles/Rxnfiles v2000 and v3000, SDF, RDF, CML, SMILES, SMARTS.
- Portability: Pre-built binary packages are provided for Linux and Windows (both 32-bit and 64-bit), and also for Mac OS X systems (both 10.5 and 10.6).
- Molecule and reaction rendering. Best picture quality among all available products. Easy SVG support.
- Automatic layout for SMILES-represented molecules and reactions.
- Canonical (isomeric) SMILES computation.
- Exact matching, substructure matching, SMARTS matching.
- Matching of tautomers and resonance structures.
- Molecule fingerprinting, molecule similarity computation.
- Fast enumeration of SSSR rings, subtrees, and edge sugraphs.
- Molecular weight, molecular formula computation.
- R-Group deconvolution and scaffold detection. Pioneer work in computing the exact maximum common substructure for an arbitrary amount of input structures.
- Combinatorial chemistry
- Plugins support in the API. As a reference, please see the Renderer plugin distributed together with the Indigo API.

Beside its portability this library contains a lot of APIs that could be very useful in the EPA ToxCast project especially for the similarity measures and chemical compounds descriptor generation. Also it's supposed to be more efficient than others as its core algorithms had been written in C.

Programming Language

C, C++, C# Java, Python

Example

Accessing Atoms and Bonds:

The following methods can be applied to a molecule or query molecule:

- getAtom — returns the atom by the given index.
- getBond — returns the bond by the given index.
- iterateAtoms — returns an iterator over atoms, including pseudoatoms and R-sites.
- iteratePseudoatoms — returns an iterator over pseudoatoms.
- iterateRSites — returns an iterator over R-sites.
- iterateBonds — returns an iterator over bonds.

Canonical SMILES:

IndigoObject.canonicalSmiles method computes the canonical SMILES (also known as absolute SMILES) string for a molecule.

Java:

```
System.out.println(mol2.canonicalSmiles());
```

C#:

```
System.Console.WriteLine(mol2.canonicalSmiles());
```

Python:

```
print mol2.canonicalSmiles()
```

Saving Molecules:

IndigoObject.smiles, when applied to a molecule, returns a SMILES string. Similarly**, IndigoObject.molfile** returns a string with a Molfile, while IndigoObject.cml returns a string with CML representation. **IndigoObject.saveMolfile** and **IndigoObject.saveCml** methods save Molfile and CML to disk**.**

Java:

```
System.out.println(mol1.molfile());
System.out.println(mol2.smiles());
qmol1.saveMolfile("query.mol");
```

C#:

```
System.Console.WriteLine(mol1.molfile());
System.Console.WriteLine(mol2.smiles());
qmol1.saveMolfile("query.mol");
```

Python:

```
print mol1.molfile()
print mol2.smiles()
qmol1.saveMolfile("query.mol")
```

For more examples check this page.

Input/output

The input format is detected automatically, except for SMARTS expressions, for which there are special methods. The input file will have (.mol) extension.

The output format depends on the operation we are going to apply on the molecules. It could be an image (png, jpg…etc.) for the rendering operations.

For molecules IndigoObject.smiles, when applied to a molecule, returns a SMILES string.
Similarly, IndigoObject.molfile returns a string with a Molfile, while IndigoObject.cml returns a string with CML representation. IndigoObject.saveMolfile and IndigoObject.saveCml methods save Molfile and CML to disk.

For more details about input and output of this toolkit please check this page.

License

This program is free software: You can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; version 3 of the License.

How to get it

All download links can be found here.

Documentation

To understand Indigo's scope, start with the Concepts page. A quick reference of the API for all supported languages is provided in the API page. A separate page deals with the C interface. Various options that can be passed to the library are explained on the Options page.

Installation

Installation for (Windows, Linux and Mac) can be found here.

Dragon

Summary

Dragon 6 is an application for the calculation of molecular descriptors. These descriptors can be used to evaluate molecular structure-activity or structure-property relationships, as well as for similarity analysis and high-throughput screening of molecule databases. Actually Dragon is widely used in scientific studies as well as part of several QSAR suites.

Dragon has been designed to run on both Windows and Linux systems and on a variety of computers. Dragon can run either interactively or in batch mode by a command line.

Why this may be useful

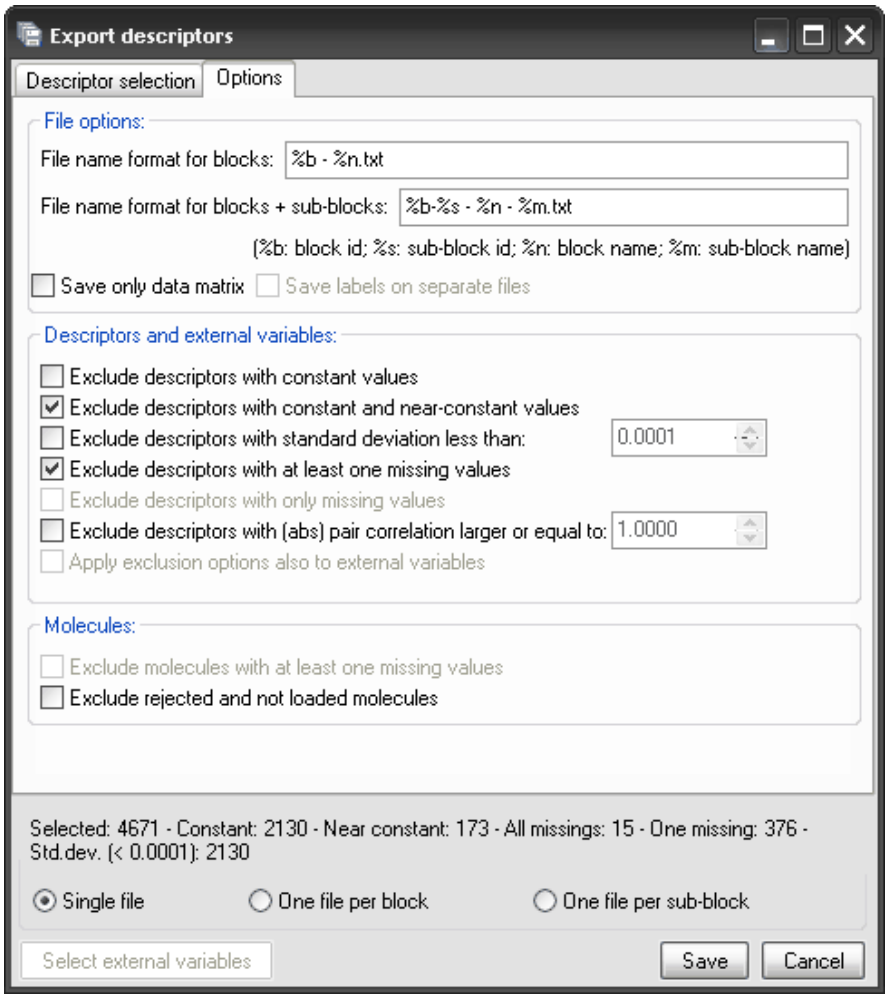
- **Descriptor calculation**. In Dragon 6.0, the user is allowed to customize descriptor calculation and choose the atom weighting schemes before descriptor calculation, whether apply logarithmic transformation to spectral moments and walk and path counts. The novel window for descriptor selection allows also selection of single molecular descriptors included in the different blocks.
- **Input files**. Together with the molecule formats allowed in the previous versions of Dragon (MDL, Sybyl, HyperChem, Macromodel and Smiles), new formats are now supported: CML (Chemical Markup Language) and HyperChem format as a unique file. There are no restrictions on the number of molecules and on the atom types. The novel Dragon graphical interface for molecule import is very flexible allowing also selection of structures that are stored by different file formats and located in different folders. Molecular structures stored in files of different types can be loaded and contemporarily processed in the same batch.
- Graphical tools are available in Dragon 6 for a preliminary descriptor analysis. These graphs allow a preliminary analysis of molecule distribution in the descriptor space, as well as a preliminary correlation analysis. **Look at the screenshots** of Dragon 6.
- Dragon can now be used inside the **KNIME environment** using the extension available **here**.
- A **molecule viewer** is provided to display the molecular structures that have been imported for descriptor calculation.
- **Loading of user-defined variables**. Dragon allows the user to import up to 200 variables and string fields that can be added to the calculated molecular descriptors such as experimental properties and information on the processed molecules (e.g., CAS number, product code).

Input/output

To calculate molecular descriptors, Dragon requires molecular structure files, which need to be previously generated by other specific chemical drawing programs. The most common molecular file formats are accepted.

To make full use of Dragon calculations, 3D optimised structures with hydrogens are required. However, Dragon can also deal with H-depleted molecules and 2D-structures; in this case, it's clear that some restrictions to descriptor calculation will be applied.

Screenshot to select the export options.



If the user decides to generate several output files, one for each selected descriptor block or sub-block, the name of the output files will be automatically generated by the program on the basis of the format specified by the user. This format can include a text file name and different tags in any position within the file name. These tags will help the user to recognize the corresponding blocks and sub-blocks in the name of the output files.

How to get it

You can order and buy Dragon (or upgrade the software from previous versions) directly on-line. There are two ways for buying Dragon on-line: you can pay directly by means of a credit card (Visa or MasterCard) or you can place your order and pay later with a bank transfer.

You can find all the details about how to buy Dragon at: http://www.talete.mi.it/order/order.htm.

Dragon prices are listed at: http://www.talete.mi.it/products/prices.php

License

Dragon is released with different commercial and academic licenses:

Single License: Dragon can be run only on the central processing unit originally designated for installation.

Site License: Dragon may be run or accessed by any computer at the specific location to which Dragon is delivered, but may not be accessed from remote sites.

Permanent License: the term of the license is perpetual. For a period of one year from software delivery, Talete srl will provide maintenance services.

Rent License: the term of the license is annual. Talete srl will provide both maintenance services and update services. Thereafter, upon payment of the renewal fee by Licensee, the term of the license will be provided for twelve-month period.

Talete srl issues academic license for non-commercial use only. Academic license is intended only for Universities. Academic license is intended only for any research from which any resulting intellectual property remains in the public domain.

Read the Dragon license agreement for further details.

Prices of all the Dragon licenses are listed in the Dragon web page: http://www.talete.mi.it/products/dragon_description.htm

Documentation

Dragon 6 user manual.

Installation

Find the installation details here under About **Dragon->Dragon Installation**.

Citation

List of citations for this library could be found here.

OEChem

Summary

OEChem is a programming library for chemistry and cheminformatics that is fast and flexible. OEChem TK has many simple yet powerful functions that **handle the details of working with small molecules**, as well as an expanding number of functions for dealing with proteins. **High-level functions** provide simplicity while **low-level functions** provide flexibility.

Why this may be useful

Here is a list of features this library provides:

- Facile management of molecules, atoms, bonds, and conformers
- Conformational and frame-of-reference coordinate transformations
- Maximum common substructure
- Substructure searching based on SMARTS or MDL query
- Perception of aromaticity with multiple models
- Chemical reaction parsing and processing

- Library generation based on SMIRKS or MDL reaction
- Tetrahedral and E/Z stereochemistry recognition
- CIP atom and bond stereo perception
- Ring perception and Kekulization
- Molecular canonicalization
- Multiconformer molecule handling
- Ability to store and recall generic primitives or user-defined objects on molecules, atoms, bonds, or conformers

One advantage of robust multiple chemistry perception is data integrity: OEChem TK is able to navigate through file formats with no loss of information. The table in the next section shows the common molecule file formats supported by OEChem.

The OEChem TK also includes two sub-libraries designed to handle macromolecules (OEBio) and grids (OEGrid).

Key features of OEBio:

- protein residue, the primary, secondary and tertiary structure hierarchy perception
- crystal symmetry handling
- sequence alignment
- management of torsions, rotamer libraries, and alternate conformations

Key features of OEGrid:

- support for the following grid file formats: Grasp, GRD (OpenEye Binary format), CCP4, XPLOR.
- Also it supports a variety of programming language.

Input/output

File formats supported by OEChem:

File Format	read	write
OpenEye's binary	Yes	Yes
MDL Mol	Yes	Yes
MDL SD	Yes	Yes
MDL RDF	Yes	No
Protein Databank PDB	Yes	Yes
Tripos Sybyl mol2	Yes	Yes
Canonical SMILES	Yes	Yes
Canonical isomeric SMILES	Yes	Yes
InChI	No	Yes
InChIKey	No	Yes
FASTA protein sequence	Yes	Yes
Macromodel	Yes	Yes
XMol XYZ	Yes	Yes

For more details about these formats please check the (input/output file formats) section you can find some sample files there.

Programming language

Java, C++, C# and python

Example

Creating a molecule from a SMILES string (C#):

```
using System;
using OpenEye.OEChem;

public class CreateOEGraphMolFromSMILES
{
    public static void Main(string[] argv)
    {
        // create a new molecule
        OEGraphMol mol = new OEGraphMol();

        // convert the SMILES string into a molecule
        OEChem.OEParseSmiles(mol, "c1ccccc1");
    }
}
```

The **OEParseSmiles** function returns a boolean value indicating whether the input string was a valid SMILES representation of a molecule.

Reading and writing molecule files(Java):

```
package openeye.docexamples.oechem;

import openeye.oechem.*;

public class ReadWriteToFiles {
    public static void main(String argv[]) {
        oemolistream ifs = new oemolistream();
        oemolostream ofs = new oemolostream();

        if (!ifs.open("input.sdf"))
            oechem.OEThrow.Fatal("Unable to open 'input.sdf'");

        if (!ofs.open("output.mol2"))
            oechem.OEThrow.Fatal("Unable to create 'output.mol2'");

        OEGraphMol mol = new OEGraphMol();
        while (oechem.OEReadMolecule(ifs, mol))
            oechem.OEWriteMolecule(ofs, mol);
        ofs.close();
        ifs.close();
    }
}
```

One convenient feature of the **open** method of **oemolstreams** is that it sets the file format associated with the stream from the file extension of the filename used as an argument.

For more examples please check links in the next section.

Documentation

- Java APIs documentation.
- C#APIs documentation.
- C++ APIs documentation.
- Python APIs documentation.

Installation

- Installation for Java
- Installation for C#
- Installation for C++
- Installation for python

Citation

1. Optimizing Fragment and Scaffold Docking by Use of Molecular Interaction Fingerprints Gilles Marcou, Didier Rognan. *J. Chem. Inf. Model.*, **2007**, *47 (1)*, 195-207.
2. Database Clustering with a Combination of Fingerprint and Maximum Common Substructure MethodsMartin Stahl, Harald Mauser. *J. Chem. Inf. Model.*, **2005**, *45 (3)*, 542-548.

License

A license file from OpenEye Scientific Software is required to run any OpenEye application. A license file can be requested/obtained by contacting OpenEye at business@eyesopen.com.

More details about license can be found here.

MolProp

Summary

The MolProp provides a customizable framework for molecular property calculation geared towards enabling rapid database filtering. Filtering attempts to eliminate inappropriate or undesirable compounds from a large set before beginning to use them in modelling studies.

Why this may be useful

Filtering attempts to eliminate inappropriate or undesirable compounds from a large set before beginning to use them in modeling studies. The goal is to remove all of the compounds that are inappropriate or undesirable. To match this need, FILTER's default filter encapsulates many of the standard filtering principles, such as removal of unstable, reactive, and toxic moieties. In addition, MolProp allows the customization of the filtering criteria to fit specific needs. The criteria for passing or failing a given molecule fall into three categories:

Physical properties

- Molecular weight
- Topological polar surface area (TPSA)
- logP
- Bioavailability

Atomic and functional group content

- Absolute and relative content of heteroatoms
- Limits on a very wide variety of functional groups

Molecular graph topology

- Number and size of ring systems
- Flexibility of the molecule
- Size and shape of non-ring chains

Beyond the standard molecular properties available from OEChem TK such as molecular weight and atom type counts,**MolProp TK** calculates XlogP [1], XlogS, and PSA [2]. There are also a variety of **ADME filter** available such as Lipinski [3], Egan [4], Veber [5] (all references can be found in citation section). In addition to calculating properties and filtering on those properties, **MolProp** provides a variety of preprocessing tools for metal and salt removal, pKa normalization, normalization, reagent selection and type checking.

Input/output

All of the data MolProp TK generates in filtering molecules can be written to a tab-separated file for easy import into a spreadsheet. This function allows for combining the values dynamically for a variety of purposes, including, but not limited to, determining which filter values best fit each project's needs.

Example

Check the Filtering Theory here.

Programming language

Java, C++, C# and python

Documentation

- Java APIs documentation.
- C#APIs documentation.
- C++ APIs documentation.
- Python APIs documentation.

Citation

1. **A New Atom-Additive Method for Calculating Partition Coefficients** Wang, R., Ying, F., Lai, L.J., *Chem. Info. Comput. Sci.*, **1997**, *37*, 615.
2. **Fast calculation of molecular polar surface area as a sum of fragment-based contributions and its application to the prediction of drug transport properties** Ertl, P., Rohde, B., Selzer, P., *J. Med. Chem.*, **2000**,*37*, 3714.
3. **Prediction of drug absorption using multivariate statistics** Egan, W.J., Merz, K.M., Baldwin, J.J., *J. Med. Chem.*,**2000**, *43*, 3867.
4. **Molecular properties that influence the oral bioavailability of drug candidates** Veber, D.F., Johnson, S.R., Cheng, H.Y., Smith, B.R., Ward, K.W., Kipple, K.D., *J. Med. Chem.*, **2002**, *45*, 2615.
5. **A bioavailability score** Martin, Y.C., *J. Med. Chem.*, **2005**, *48*, 3164.

License

A license file from OpenEye Scientific Software is required to run any OpenEye application. A license file can be requested/obtained by contacting OpenEye at business@eyesopen.com.

More details about license can be found here.

GraphSim

Summary

Measuring molecular similarity and diversity plays an important role in various steps of the drug design cycle. Calculating molecule similarity is extensively used in applications such as virtual screening, property prediction, synthesis design and chemical database clustering.

Why this may be useful

This library has the ability to perform similarity measures between molecules. Fingerprinting provides an elementary encoding of molecular graphs. Even though fingerprints can only represent local structural features and not their relative positions in molecules, it has proven to be very successful in a range of similarity and diversity studies.

GraphSim provides five different fingerprint types to perform 2D molecular similarity measurements:

- Path
- Circular [1]
- Tree
- MACCS key [2]
- LINGO [3]

Also GraphSim supports several built-in similarity coefficients (*Cosine*, *Dice*, *Euclidean*, *Manhattan*, *Tanimoto*, *Tversky*), while user-defined similarity measures are also available.

Apart from generating and storing fingerprints, **GraphSim** also provides a fingerprint database that is designed to perform rapid in-memory fingerprint search utilizing any of the built-in or user-defined similarity measures.

GraphSim also provides access to the common fragments found between two molecules based on the given fingerprint type.

Input/output

Input supposed to be molecular files for and the output the fingerprint descriptors, or similarity calculations.

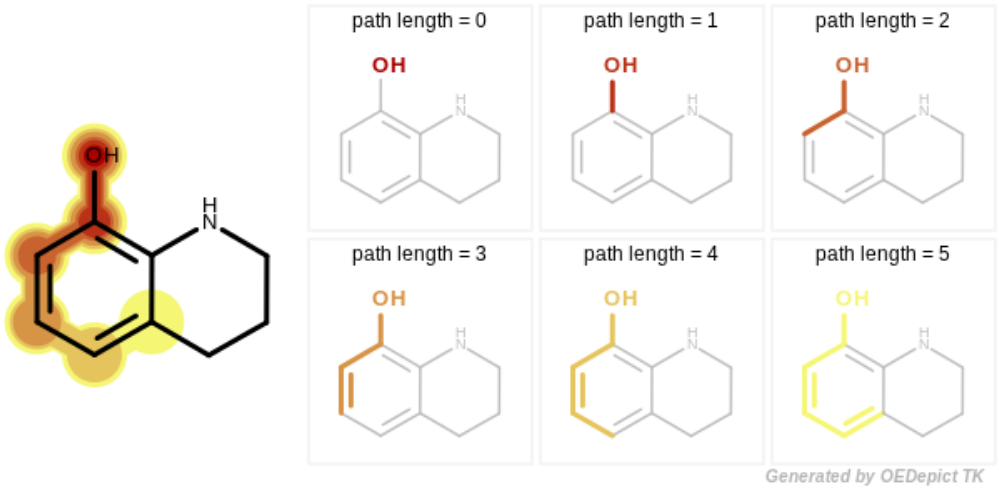
Also we can perform a search on a database. In this case the input will be a query and the output supposed to be a search result.

Programming language

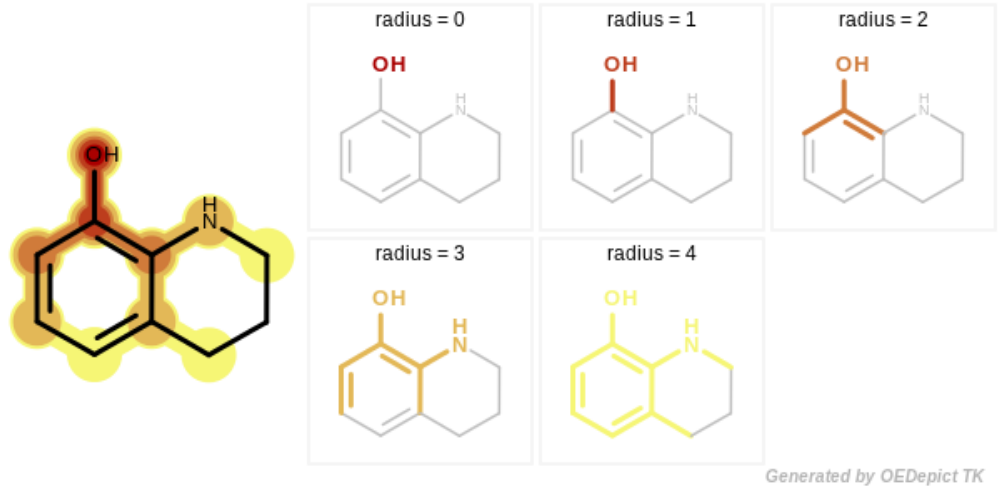
C++, C#, Java and Python.

Example

A *path fingerprint* is generated by exhaustively enumerating all linear fragments of a molecular graph up to a given size and the hashing these fragments into a fixed-length bivector.



A *circular fingerprint* is generated by exhaustively enumerating all circular fragments grown radially from each heavy atom of the molecule up to the given radius.



Documentation

- Java APIs documentation.
- C#APIs documentation.
- C++ APIs documentation.
- Python APIs documentation.

Citation

1. **Extended-Connectivity Fingerprints**, D. Rogers, M. Hahn. *J. Chem. Inf. Model.*, **2010**, *50*, (5) 742-754.
2. **Reoptimization of MDL Keys for Use in Drug Discovery**, J. L. Durant, B. A. Leland, D. R. Henry, J. G. Nourse. *J. Chem. Inf. Comput. Sci.*, **2002**, *42*, (6) 1273-1280.
3. **LINGO, an Efficient Holographic Text Based Method To Calculate Biophysical Properties and Intermolecular Similarities**, D. Vidal, M. Thormann, M. Pons. *J. Chem. Inf. Model.*, **2005**, *45*, (2) 386-393.
4. **[IMPORTAN] In vitro and in silico affinity fingerprints: Finding similarities beyond structural classes**, H. Briem, U. F. Lessel. *Perspectives in Drug Discovery and Design*, **2004**, *20*, 231-244.
5. **Comparison of Fingerprint-Based Methods for Virtual Screening Using Multiple Bioactive Reference Structures**, J. Hert, P. Willett, D. J. Wilton. *J. Chem. Inf. Comput. Sci.*, **2004**, *44*, (3) 1177-1185.

License

A license file from OpenEye Scientific Software is required to run any OpenEye application. A license file can be requested/obtained by contacting OpenEye at business@eyesopen.com.

More details about license can be found here.

MolEngine

Summary

MolEngine is a complete .NET Cheminformatics Toolkit completely built on Microsoft .NET platform. By using MolEngine, developers can build variety of applications on multiple platforms:

- ASP.NET Web Application
- Click-once Applicaton
- Office Add-on
- SharePoint Application
- Windows 8 Metro App
- Silverlight Application
- MVC Application
- Native Application on Linux using Mono
- ASP.NET Web Application on Linux using Mono + XSP2

MolEngine enables .NET developer to quickly build chemistry desktop and web applications without dealing with the complicated Cheminformatics algorithms.

MolEngine is great for Window Azure platform to implement chemical structure search. Here is an example

Why this may be useful

Here is a summary of functionalities of MolEngine:

- Read and write all popular chemistry files including CDX, CDXML, SKC, Molfile, SDF, Mol2, CML, MRV, RXN, RDF, SMILES, InChI, TGF etc.
- Generate 2D coordinates
- Combinatorial Chemistry enumeration
- Generate Structure Fingerprints
- Reaction mapping
- Structure search, including substructure, full-structure, and similarity
- Generate structure images
- Generate structure hash code
- Convert spectrum files (JDX, JD) into images

Programming language

MolEngine library provide APIs for .Net languages whoever there are a MolEngine Web services which warps MolEngine APIs as SOAP web service, which can be easily called from remote computers by other languages such as Java, objective-c, c++, python …etc.

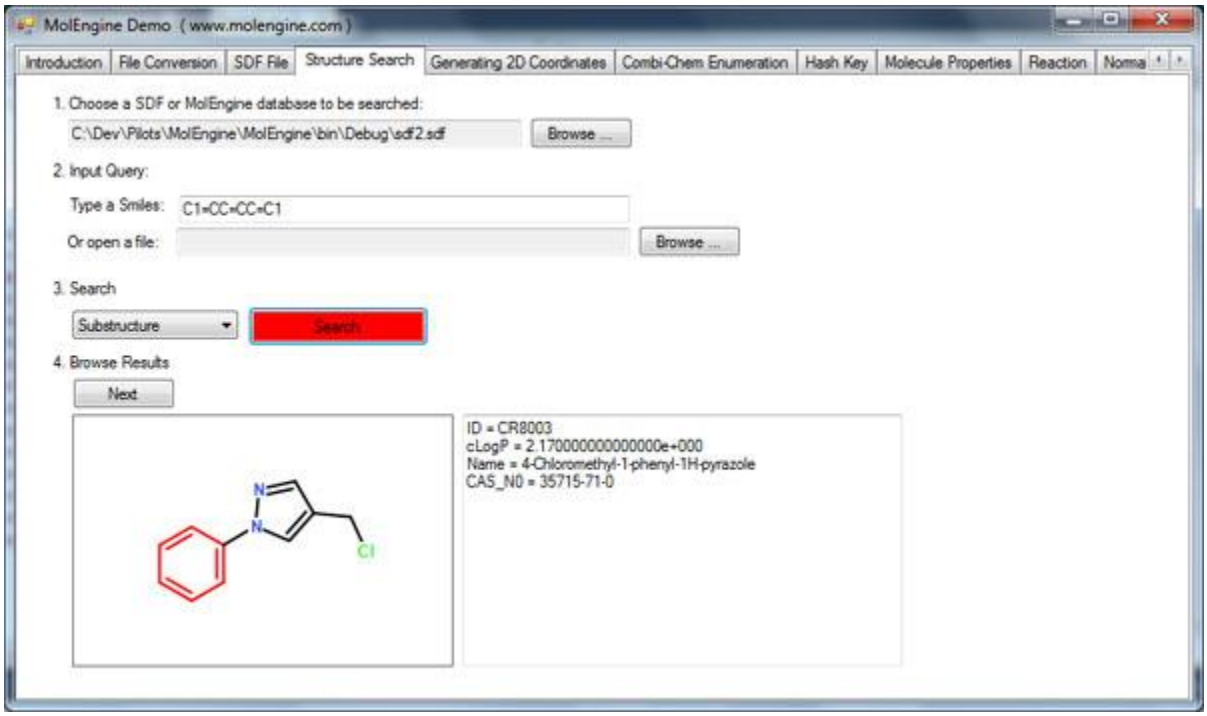
Input/output

Reading and writing all popular chemistry CDX, CDXML, SKC, Molfile, SDF, Mol2, CML, MRV, RXN, RDF, SMILES, InChI, TGF etc.

For more details about each format you can check (Input/output file formats) section.

Example

Here is a screen shot from a sample application using the MolEninge APIs.



Documentation

- MolEngine APIs documentation
- MolEngine Web Services documentation

How to get it

You can download it from here

License

Evaluation license agreement details here.

MARVIN Beans

Summary

Marvin is a collection of tools for drawing, displaying and characterizing chemical structures, queries, macromolecules and reactions. The Calculator Plugins, included in the Marvin Suite, are modules of ChemAxon's Marvin and JChem cheminformatics platforms which calculate chemical properties descriptors from chemical structures. An extension for KNIME is also available.

ChemAxon's Calculators and Calculator Plugin suite is a cheminformatics toolkit built on integrated technologies that achieve unmatched performance and versatility. It offers solutions to a wide range of problems faced by researchers with all levels of modeling expertise. Implemented calculations and property predictions efficiently evaluate pharmaceutically relevant physico-chemical properties and molecular descriptors even for hundreds of thousands compounds, making it a powerful lead generation and lead optimization tool.

Why this may be useful

This tool is very reliable. It had been used in a lot of publications. Beside that It provide all needed functionality like calculating descriptors(provide over 400 descriptors), performing substructure searching, duplicate structure search and exact structure searching, searching in database and other searches. Moreover it has the ability to be integrated with a variety of database systems (Oracle, MS SQL Server, DB2, Access, etc.).

Input/output

This tool provides both GUI and APIs (in .Net, java and JavaScript).

The input supposed to be molecules files or a molecules database. Output will be the result of different operations supported by Marven toolkit.

The file formats in Marven please check this page.

Example

Duplicate Structure Search:

This search type can be used to retrieve the same molecule as the query. It is used to check whether a chemical structure already exists in the database, and also during duplicate filters import. All structural features (atom types, isotopes, stereochemistry, query features, etc.) must be the same for matching, but for example coordinates and dimensionality are usually ignored

Java example: Throwing an exception if a given structure exists.

```
...                                    // Initialize connection
String mol = "Clc1cccc(Br)c1"; // Query in SMILES/SMARTS, MDL Molfile or other format
String structureTableName = "cduser.structures";
JChemSearch searcher = new JChemSearch(); // Create searcher object
searcher.setQueryStructure(mol);
searcher.setConnectionHandler(connHandler);
searcher.setStructureTable(structureTableName);
JChemSearcOptions searchOptions = new JChemSearchOptions(SearchConstants.DUPLICATE);
searcher.setSearchOptions(searchOptions);
searcher.run();
if (searcher.getResultCount() > 0) {
    System.out.println("Structure already exists (cd_id=" + searcher.getResult(0) + ")");
}
```

Starting a Search

The initialization of substructure searching is similar to duplicate searching, but the JChemSearchOptions object needs to be created with SearchConstants.SUBSTRUCTURE constant value.

Java example:

```
...                                    // Initialize connection
String mol = "[*]c1cccc([Cl,Br])c1"; // Query structure
String structureTableName = "cduser.structures";
JChemSearch searcher = new JChemSearch(); // Create searcher object
```

For more examples please check the developer guide page.

How to get it

Through download page. Just choose your preferences, accept the license terms and click Download Marvin Beans after registration.

Documentation

- .Net APIs documentation
- Java APIs documentation

License

For the license details check this page.

Citations

There is a set of publications that used this tool. You can find it here.

ADRIANA.Code

Summary

ADRIANA.Code comprises a unique combination of methods for calculating molecular structure descriptors on a sound geometric and physicochemical basis. These descriptors can be used for a wide range of applications in all areas of chemistry, in particular in drug design. Lead discovery and optimization, diversity assessment of compound libraries and prediction of ADME/Tox properties are some of the problems that have been addressed and successfully solved with descriptors from ADRIANA.Code.

In addition, the descriptors have been used to model chemical reactivity, the scope and limitation of chemical reactions and to simulate spectra for structure elucidation.

Why this may be useful

ADRIANA.Code provides machinery for the representation of molecular structures. The user can bring into consideration her or his knowledge on the types of effects that are influencing the physical, chemical or biological property to be modeled. The more the user has knowledge about those effects and their influence on the property, the better a guided choice on the types of descriptors that should be used can be made.

The descriptors calculated by ADRIANA.Code can be used for the prediction of a variety of physical, chemical or biological properties that cannot be directly calculated from first principles.

Here is a set of features in this software:

Descriptors

- Physicochemical properties (global descriptors) including number of H bond acceptors and H bond donors, logP, logS, TPSA, molecular weight, dipole moment, polarizability, number of Lipinski Ro5 violations, ring and molecular complexity
- Shape- and size-related descriptors including molecular diameter, principal moments of inertia,molecular span, radius of gyration, molecular eccentricity and asphericity
- Autocorrelation of 2D interatomic distance distributions weighted by partial atom charges, electronegativities and polarizabilities
- Autocorrelation of 3D interatomic distance distributions weighted by partial atom charges, electronegativities and polarizabilities
- Radial distribution functions (RDF) of 3D interatomic distances weighted by partial atom charges, electronegativities and polarizabilities
- Autocorrelation of distances between surface points weighted by molecular electrostatic potential, hydrogen bonding potential and hydrophobicity potential

Speed and Reliability:

- Handles properly any organic compound
- Processes a data set of 100,100 small to medium-sized molecules in 1.7 h on a 1.0 GHz workstation (all global descriptors, 62 ms/cpd, 99.8% conversion rate)

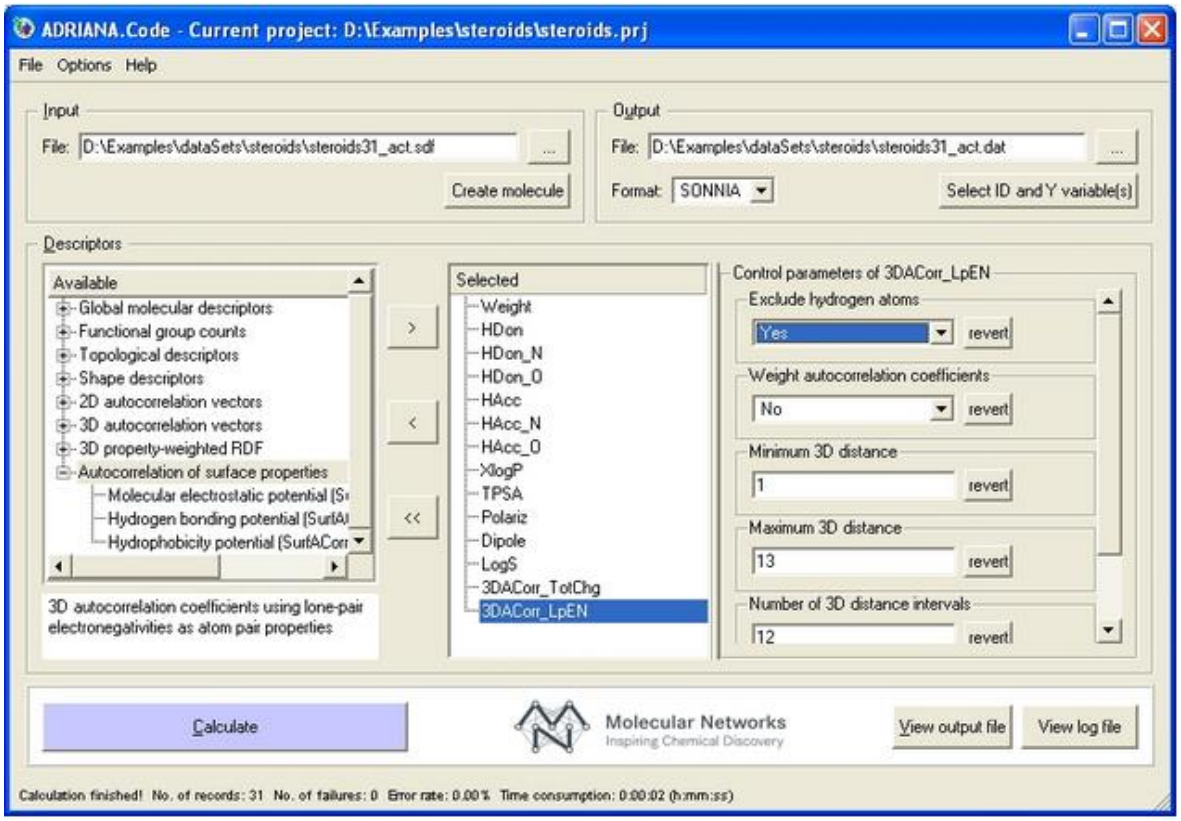
Input/output

This software provides three kinds of user interfaces:

- Graphical user interface.
- Command line interface supporting batch mode.

The current version of ADRIANA.Code supports MDL SDFile, SMILES linear notation and Gasteiger ClearText (CTX) file formats for structure input.

The output formats supported in the current version of ADRIANA.Code are MDL SDF, TSV(tab separated value), CSV(comma separated value) and SONNIA data input file.



For examples on these file formats please check the (Input/Output file formats) section in this document.

How to get it

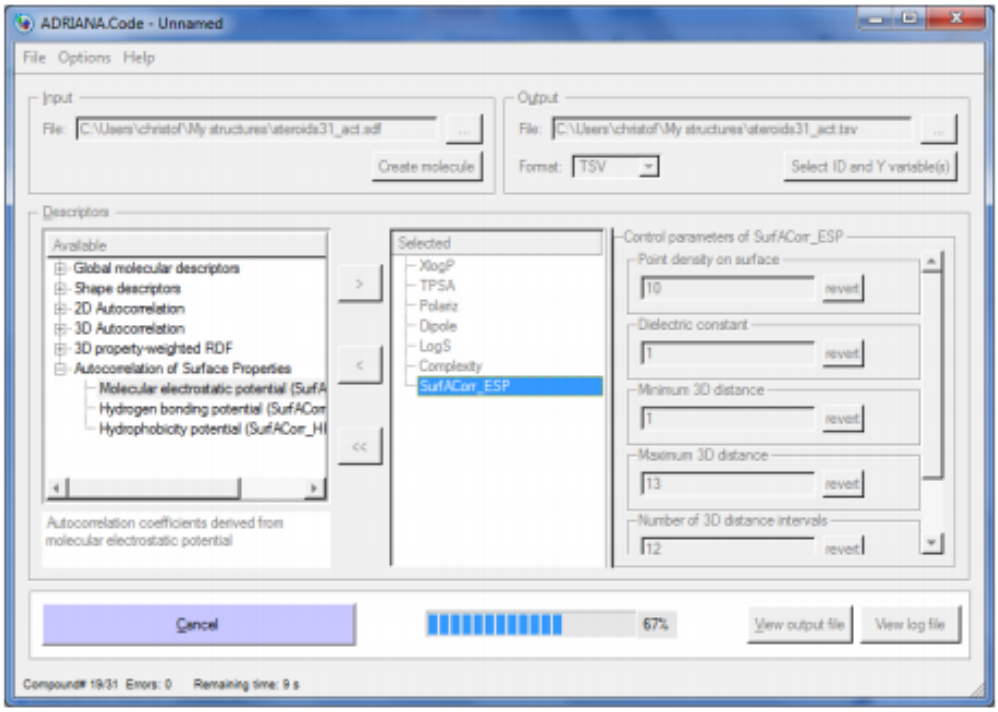
Evaluation Version

A demo version is available on request free of charge. Please register in our Download Area.
Note: for evaluation version you need to register with an academic institution or company email address.
For full version you can contact them. All contact info can be found here. Pricing is not available on their website.

Example

Descriptors of ADRIANA.Code

The Descriptors section in the middle part of the ADRIANA.Code GUI consists of three different areas. The left list view Available displays all descriptors that can be calculated with ADRIANA.Code. The list view Selected in the middle shows all descriptors that are selected by the user to be calculated. In the right area Available Control Parameters all control parameters that are available for a certain descriptor are displayed for editing in this section. All available control parameter have preset (default) values. Note. Some descriptors do not have any control parameters (e.g., molecular dipole moment, topological polar surface area or mean molecular polarizability).



For full description of the descriptors check this page.

Documentation

Adriana.Code user Manuel

ADRIANA.Code and SONNIA tutorial

License

Software Licensing Agreement of Molecular Networks GmbH Computerchemie. For more info about the license terms you may have to contact them.

Citations

Here is a list of publications used this technology. **This publication more related to the EPA ToxCost project.**

Selected Application Areas:

- Analysis of HTS results
- Prediction of the isoform specificity of human CYP450 substrates
- Finding new lead structures and lead hopping
- Modeling biological activities

Protégé

Summary

The Protégé library represents the approach of using description logics and ontology-based approach to represent description of classes (in our case the compounds of chemicals), their properties and concrete instances. Given the description logic semantics we can encode known chemical structures and infer new knowledge given the base knowledge we have and inference rules.

Why this may be useful

The competitors can define rules and querying the knowledge base to infer similarity measure and output the solution to similarity problems. There is already an ongoing usage of the approach in EPA [1], [2]. There is broad field of representing chemicals with ontologies that have prominent results [3]. As the field is established it is expected that positive results can definitely be derived using this approach.

Here is a list of features this library provides:

- Provides a way to encode known chemical structures and infer new knowledge.
- Open source.
- Extensible.
- Provides a plug-and-play environment that makes it a flexible base for rapid prototyping and application development.

Example

Suppose the following rule is known: "If there is a set of 6 carbon atoms, and each i-th atom is connected to (i+1) – th atom in the set, then every atom in the set is called ring atom". This can be shown as following rule:

CarbonAtom(x) ^ CarbonAtom(y) ^ CarbonAtom(z) ^ CarbonAtom(w) ^

CarbonAtom(u) ^ CarbonAtom(v) ^ hasBondWith(x,y) ^

hasBondWith(y,z) ^ hasBondWith(z,w) ^ hasBondWith(w,u) ^

hasBondWith(u,v) ^ hasBondWith(v,x) → RingAtom (x)

Where CarbonAtom(a) specifies that atom a is Carbon, hasBondWith(a,b) specifies that atoms a and b are connected.

Input/output

The input to the library is the set of encoded facts and rules. (See the previous example).

The output from the library is the facts that can be inferred from knowledge. In case there are six atoms a,b,c,d,e,f are all carbons and connected we can deduce that any of this atom set is also an ring atom.

How to get it

The library can be downloaded directly here.

Documentation

You can check this wiki page for lots of documentations

Limitation

There is a big risk and limitations which comprise that competitors should build their solutions mostly from pure skretch with no real framework/workspace to work with chemical components. Additionally, there would an additional effort for integrating it into TopCoder platform.

License

This library is free and open source. For more details about the license check this page.

Citations

[1] - http://www.epa.gov/oei/symposium/2008/raskin.pdf

[2] - http://www.epa.gov/ncct/expocast/files/data/PRoTEGE_CCL_Ranking__all-chemicals-report-draft_2011-04-07.pdf

[3] - http://researchspace.csir.co.za/dspace/bitstream/10204/4919/1/Britz1_2010.pdf

---

Libraries Summary

Name	License	APIs	Home Page
Chemistry Development Kit(CDK)	Open Source	Java	http://sourceforge.net/projects/cdk/
Open Babel	Open source	C++,Python, Ruby	http://openbabel.org/wiki/Main_Page
OpenTox	Open source	Web-Services REST APIs	http://www.opentox.org
ChemmineR	Open source	R	http://manuals.bioinformatics.ucr.edu/home/chemminer
RDKit	Open source	C++, python	http://www.rdkit.org/
Indigo	Open source	C, C++, C#, Java, Python	http://ggasoftware.com/opensource/indigo
Dragon 6	Proprietary/Free for Academic	GUI/Command Line Tool	http://www.talete.mi.it/dragon.htm
OEChm	Proprietary	C++, C#, java and python	http://www.eyesopen.com/oechem-tk
MolProp	Proprietary	C++, C#, java and python	http://www.eyesopen.com/molprop-tk
GraphSim	Proprietary	C++, C#, java and python	http://www.eyesopen.com/graphsim-tk
MolEngine	Proprietary/Evaluation version available	.Net APIs/Web-Services	http://www.scilligence.com/web/molengine.aspx
Marvin Beans	Proprietary/free for academic	Java, .Net and Javascript	http://www.chemaxon.com/
Adriana.Code	Proprietary/ Evaluation version available by registration	GUI/Command Line	http://www.molecular-networks.com/products/adrianacode
Protégé	Open source	Encoded facts and rules	http://protege.stanford.edu/

Input/output File Formats

In this section I will present a table which contains different input and output formats with a sample file shows the format for each type.

Type	Filename extension	Description	Example File	Source of Information/Display Program
chemical/x-alchemy	alc	Alchemy format	example.alc	http://www.camsoft.com/plugins/
chemical/x-cache-csf	csf		example.csf	
chemical/x-cactvs-binary	cbin	CACTVS binary format	example.cbin	http://cactvs.cit.nih.gov
chemical/x-cactvs-binary	cascii	CACTVS ascii format	example.cascii	http://cactvs.cit.nih.gov
chemical/x-cactvs-binary	ctab	CACTVS table format	example.ctab	http://cactvs.cit.nih.gov
chemical/x-cdx	cdx	ChemDraw eXchange file	example.cdx	http://www.camsoft.com/plugins/
chemical/x-cerius	cer	MSI Cerius II format	example.cer	http://www.msi.com/
chemical/x-chemdraw	chm	ChemDraw file	example.chm	http://www.camsoft.com/plugins/
chemical/x-cif	cif	Crystallographic Interchange Format	example.cif	http://www.bernstein-plus-sons.com/software/rasmol/ http://ndbserver.rutgers.edu/NDB/mmcif/examples/index.html
chemical/x-mmcif	mcif	MacroMolecular CIF	example.mcif	http://www.bernstein-plus-sons.com/software/rasmol/ http://ndbserver.rutgers.edu/

				NDB/mmcif/examples/index.html
chemical/x-chem3d	c3d	Chem3D Format	example.c3d	CambridgeSoft
chemical/x-cmdf	cmdf	CrystalMaker Data format	example.cmdf	http://www.crystalmaker.co.uk/
chemical/x-compass	cpa	Compass program of the Takahashi	example.cpa	
chemical/x-crossfire	bsd	Crossfire file	example.bsd	
chemical/x-cml	cml	Chemical Markup Language	example.cml	http://cml.sourceforge.net/
chemical/x-csml	csml, csm	Chemical Style Markup Language	example.csml	http://www.mdli.com/
chemical/x-ctx	ctx	Gasteiger group CTX file format	example.ctx	
chemical/x-cxf	cxf		example.cxf	
chemical/x-daylight-smiles	smi	Smiles Format	example.smi	http://www.daylight/dayhtml/smiles/index.html
chemical/x-embl-dl-nucleotide	emb	EMBL nucleotide format	example.emb	http://mercury.ebi.ac.uk/
chemical/x-galactic-spc	spc	SPC format for spectral and chromatographic data.	example.spc	http://www.galactic.com/galactic/Data/spcvue.htm
chemical/x-gamess-input	inp, gam	GAMESS Input format	example.inp	http://www.msg.ameslab.gov/GAMESS/Graphics/MacMolPlt.shtml
chemical/x-gaussian-input	gau	Gaussian Input format	example.gau	http://www.mdli.com/
chemical/x-gaussian-checkpoint	fch,fchk	Gaussian Checkpoint format	example.fch	http://products.camsoft.com/
chemical/x-gaussian-cube	cub	Gaussian Cube (Wavefunction) format	example.cub	http://www.mdli.com/

chemical/x-gcg8-sequence	gcg		example.gcg	
chemical/x-genbank	gen	ToGenBank format	example.gen	http://www2.ebi.ac.uk:80/egcg-html/
chemical/x-isostar	istr, ist	IsoStar Library of intermolecular interactions	example.istr	http://www.ccdc.cam.ac.uk/
chemical/x-jcamp-dx	jdx, dx	JCAMP Spectroscopic Data Exchange Format	example.dx	http://www.mdli.com/
chemical/x-kinemage	kin	Kinetic (Protein Structure) Images	example.kin	http://www.faseb.org/protein/kinemages/MageSoftware.html
chemical/x-macmolecule	mcm	MacMolecule File Format	example.mcm	http://www.molvent.com/
chemical/x-macromodel-input	mmd, mmod	MacroModel Molecular Mechanics	example.mmd	http://www.columbia.edu/cu/chemistry/mmod/mmod.html
chemical/x-mdl-molfile	mol	MDL Molfile	example.mol	http://www.mdli.com/
chemical/x-mdl-rdfile	rd	Reaction-data file	example.rd	http://www.mdli.com/
chemical/x-mdl-rxnfile	rxn	MDL Reaction format	example.rxn	http://www.mdli.com/
chemical/x-mdl-sdfile	sd	MDL Structure-data file	example.sd	http://www.mdli.com/
chemical/x-mdl-tgf	tgf	MDL Transportable Graphics Format	example.tgf	http://www.mdli.com/
chemical/x-mif	mif		example.mif	
chemical/x-mol2	mol2	Portable representation of a SYBYL molecule	example.mol2	http://www.tripos.com/TechBriefs/mol2_format/mol2.html
chemical/x-molconn-Z	b	Molconn-Z format	example.b	http://www.eslc.vabiotech.com/molconn/molconnz.html
chemical/x-mopac-input	mop	MOPAC Input format	example.mop	http://www.mdli.com/

chemical/x-mopac-graph	gpt	MOPAC Graph format	example.gpt	http://products.camsoft.com/
chemical/x-ncbi-asn1	asn (old form)		example.asn	
chemical/x-ncbi-asn1-binary	val		example.val	
chemical/x-pdb	pdb	Protein DataBank	example.pdb	http://www.mdli.com/
chemical/x-swissprot	sw	SWISS-PROT protein sequence database	example.sw	http://www.expasy.ch/spdbv/text/download.htm
chemical/x-vamas-iso14976	vms	Versailles Agreement on Materials and Standards	example.vms	http://www.acolyte co uk/JISO/
chemical/x-vmd	vmd	Visual Molecular Dynamics	example.vmd	http://www.ks.uiuc.edu/Research/vmd/
chemical/x-xtel	xtel	Xtelplot file format	example.xtel	http://www.iumsc.indiana.edu/graphics/xtelplot/xtelplot.htm
chemical/x-xyz	xyz	Co-ordinate Animation format	example.xyz	http://www.mdli.com/

Ideas and Approaches

Computing Toxic Chemicals

Idea inspiration

Toxicity is almost always an issue of availability and dosage. Whether or not a compound is natural or synthetic it can be toxic from snake venom and jellyfish stings to petrochemicals and pesticides. However, some chemicals are more toxic than others; exposure to a lower dose will cause health problems or potentially be lethal. It is very important to find a way to determine whether a newly discovered synthetic or natural chemical might cause toxicity problems.

The team also points out that the US Environmental Protection Agency (EPA) and the Office of Toxic Substances (OTS) in the USA had listed 70,000 industrial chemicals in the 1990s, with 1000 chemicals added each year for which even simple toxicological experiments had not been carried out. This is largely a problem of logistics and costs as well as the ethical question of whether so many tests, which would have to be carried out on laboratory animals, should be done at all.

A research team in the Department of Electrical Engineering and Computer Science at Kansas, have successfully tested a statistical algorithm against more than 300 chemicals for which the toxicity profile is already known. Their technique offers a computational method of screening a large number of compounds for obvious toxicity very quickly and might preclude the need for animal testing of the compounds, provided regulators don't insist on such "in vivo" data from the latter.

The research builds on well-established principles from the pharmaceutical industry known as Quantitative structure-activity relationships (QSARs) in which the type of atoms and how they are connected together can be correlated with the activity of a drug molecule. Certain molecular shapes and types are soluble in water, for instance, or interact in a certain way with different enzymes and other proteins in the body, leading to their overall activity. Different molecular features will make a similar molecule behave in a different way -- more or less soluble, stronger or weaker acting. The team has now turned the QSAR around so that instead of searching for the features in a molecule that make it of benefit in medicine they look for the atomic groups and the type of bonds that hold them together to find associations with toxicity.

The team points out that few earlier attempts at predicting toxicity of chemicals have proved successful, most approaches are no better than random guessing. The team's new statistical approach combines "Random Forest" selection with "Naïve Bayes" statistical analysis to boost the predictions well beyond random. They team saw prediction accuracy in 2 out of 3 chemicals tested. Given that there are around 100,000 industrial chemicals that need toxicity profiling, this result should allow the industry and regulators to focus on a large number of the most pressing of those, the ones predicted to have greatest toxicity and leave the less likely until additional resources are available.

The researchers are now tuning the algorithm to work faster and with greater precision so that it ignores common molecular features now known not to contribute to toxicity characteristics in the chemicals they have studied so far.

This research is too similar to the idea of the EPA ToxCast project. The problem is that the research paper is not available publically online. But you may find it here.

References

- Meenakshi Mishra, Hongliang Fei, Jun Huan. Computational prediction of toxicity. Int. J. Data Mining and Bioinformatics, 2013
- Post link

Easier way to make new compounds

Idea inspiration

Scientists at The Scripps Research Institute have developed a powerful new technique for manipulating the building-block molecules of organic chemistry. The technique enables chemists to add new functional molecules to previously hard-to-reach positions on existing compounds -- making it easier for them to generate new drugs and other organic chemicals. They build a basic tool for making novel chemical compounds, and it should have a wide range of applications.
The new advance is a method for "CH activation" -- chemists' code for the removal of a simple hydrogen atom from the carbon backbone of an organic molecule, and the replacement of that hydrogen atom with a functional chemical group. Compared to the traditional method, in which chemists modify only the existing functional groups on a compound, CH activation more directly boosts the complexity of a compound, giving it potentially valuable new properties.

Just a Small Sample of the Possibilities:
The team used the technique to quickly modify a variety of compounds, including the amino acid phenylalanine and the neurotransmitter-mimicking drug baclofen. These are from compound classes that chemists use routinely to synthesize new candidate drugs and other useful chemicals, and they represent just a small sample of the possibilities; they think that they can apply their technique to many compound classes and functional groups

So I think we can follow the same approach in the EPA ToxCast project where we need to discover and generate new like-toxic compounds to be used for training.

Reference

Databases

Why we need datasets:

- Calculate your molecular descriptors on the provided molecules (available formats: SMILES, HyperChem, MDL SDF).
- Eventually compare your descriptors with some well-known descriptors which are also provided in the data files.

This section will contain a list of databases for different chemical compounds descriptors.

Taxonomy database

Description

This database contains the names and phylogenetic lineages of more than 160,000 organisms that have molecular data in the NCBI databases. New taxa are added to the Taxonomy database as data are deposited for them.

Links

- Home Page

NCI Open Database Compounds

Description

These files will contain a successively curated structure set of all records of the Open NCI Database. The basis of this first file is the version of the Open NCI Database as provided by DTP.

The file was processed in the following way:

- The originally provided data fields "Release", "Structure Source" and "Structure Evaluation" were preserved.
- All name fields of the original file were merged into one data field (""DTP names")
- Addition of hydrogen atoms was performed by CACTVS.
- 3D Atom coordinates have been calculated by CORINA (if the calculation failed, 2D coordinates were calculated by CACTVS).
- Data fields "Formula" and "Molecular Weight" were added (calculated by CACTVS).
- The IUPAC Structure Identifiers "Standard InChI" and "Standard InChIKey" (Version 1.04) were included as data fields.

- NCI/CADD's Structure Identifiers "FICTS", "FICuS", and "uuuuu" were calculated and added as data fields.
- The number of potential stereo centers on atoms and/or bonds has been included as data fields "Number of atom stereocenters" and "Number of bond stereo centers"; the additional boolean field "Full atom and bond stereo specification" indicates whether full relative stereo configuration is available for the corresponding structure record (this field is missing if no stereo centers are present).

Links

- Home page
- Download Link(Release 4).

Comparative Toxicogenomics Database

Description

Files contain the current comparative toxicongenomics database.

Links

- Download Link

SET OF 2.6 MILLION UNIQUE COMPOUNDS

Description

3.8 million Of compounds from structural databases of 32 providers has been gathered and stored in our database. Once the duplicates are removed using the InChI, 2.6 million of compounds remain. The 32 databases and the whole database were studied in term of uniqueness, diversity, frameworks, drug-like and lead-like properties. This study shows that there are more than 87 000 frameworks in our database. There are 1.9 million of drug-like molecules among which more than 900 000 are lead-like. The drug likeness and lead likeness are estimated using in house scores using function to estimate convenience to properties rather than cut-off values. The compounds are stored in a MySQL database and the code to manage this database is in Java. In consequence, we have a free and easily updatable system for chemical databases management and screening sets generation.

Links

- Home Page.

LIGAND

Description

Ligand Expo (formerly Ligand Depot) provides chemical and structural information about small molecules within the structure entries of the Protein Data Bank. Tools are provided to search the PDB dictionary for chemical components, to identify structure entries containing particular small molecules, and to download the 3D structures of the small molecule components in the PDB entry. A sketch tool is also provided for building new chemical definitions from reported PDB chemical components.

Links

- Download Page

MolPort database

Description

The MolPort database of commercially available compounds is available for download in SD file format. The database is updated monthly and incremental updates are provided too. For your convenience, data has been divided by availability and application (screening compounds or building blocks).

Links

- Download Page

EcoTox Database

Description

The ECOTOXicology database (ECOTOX) is a source for locating single chemical toxicity data for aquatic life, terrestrial plants and wildlife. ECOTOX was created and is maintained by the U.S.EPA, Office of Research and Development (ORD) , and the National Health and Environmental Effects Research Laboratory's (NHEERL's) Mid-Continent Ecology Division (MED).

Links

- Home Page
- Download Page

MOLE DB

Description

The MOLE db - Molecular Descriptors Data Base is a free on-line database comprised of 1124 molecular descriptors calculated for 234773 molecules. At the present moment, 18143 queries have been made on the database. This data base is intended as a research and teaching tool.

Links

- Home Page

---

18 octane isomers (C8)

Description

The data set is constituted by 18 octane isomers (C8).

The following properties are given, both in Excel format (C8_Properties.xls) and in a tab-delimited text format (C8_Properties.txt):

- boling point (BP)
- melting point (MP)
- heat capacity at T constant (CT)
- heat capacity at P constant (CP)
- Entropy (S)
- density (DENS)
- enthalpy of vaporization (HVAP)
- standard enthalpy of vaporisation (DHVAP)
- enthalpy of formation (HFORM)
- standard enthalpy of formation (DHFORM)
- motor octane number (MON)
- molar refraction (MR)
- acentric factor (AcenFac)
- total surface area (TSA)
- octanol-water partition coefficient (LogP)
- molar volume (MV)

A set of 102 molecular descriptors (topological descriptors) is also given, both in Excel format (C8_Descriptors.xls) and in a tab-delimited text format (C8_Descriptors.txt).

Links

- Download Link

---

82 polyaromatic hydrocarbons (PAH)

Description

The data set is constituted by 82 polyaromatic hydrocarbons (PAH).

The following properties are given, both in Excel format (PAH_Properties.xls) and in a tab-delimited text format (PAH_Properties.txt):

- melting point (MP)
- boling point (BP)
- octanol-water partition coefficient (LogP)

A set of 112 molecular descriptors (topological descriptors) is also given, both in Excel format (PAH_Descriptors.xls) and in a tab-delimited text format (PAH_Descriptors.txt).

Links

- Download Link

---

209 polychlorobiphenyls (PCB)

Description

The data set is constituted by 209 polychlorobiphenyls (PCB).

The following properties are given, both in Excel format (PCB_Properties.xls) and in a tab-delimited text format (PCB_Properties.txt):

- melting point (MP)
- relative retention time (RTT)
- octanol-water partition coefficient (logP)
- total surface area (TSA)
- log Henry constant (logH)
- log water solubility (logSw)
- log water activity coefficient (logYw)
- relative enthalpy of formation (dHf)

A set of 106 molecular descriptors (topological descriptors) is also given, both in Excel format (PCB_Descriptors.xls) and in a tab-delimited text format (PCB_Descriptors.txt).

Links

- Download Link

---

22 Phenetylamines

Description

The data set is constituted by 22 phenetyl-amines with two substituent sites (Phenet).

The following property is given:

- biological activity: log(1/C)

A set of 110 molecular descriptors (topological descriptors) is also given, both in Excel format (Phenet_Descriptors.xls) and in a tab-delimited text format (Phenet_Descriptors.txt).

Links

- Download Link

Online Chemical database

Overview

OCHEM is a Free open access site of annotated models and chemical data. OCHEM represents a new way of free sharing of chemical information and knowledge on the Internet. The site allows users to publish QSPR/QSAR models with the related data on the Web - and to make them accessible and applicable to the public.

- Build QSAR models for predictions of chemical properties. The models can be based on the experimental data published in our database.
- Apply one of the available models to predict property you are interested in for your set of compounds.

Links

- Video Tutorial
- Home page

Helper Tools

The OECD QSAR Toolbox

Overview

To increase the regulatory acceptance of (Q)SAR methods, the OECD has started the development of a QSAR Toolbox to make (Q)SAR technology readily accessible, transparent, and less demanding in terms of infrastructure costs.

The Toolbox is a software application intended to be used by governments, chemical industry and other stakeholders in filling gaps in (eco)toxicity data needed for assessing the hazards of chemicals. The Toolbox incorporates information and tools from various sources into a logical workflow. Crucial to this workflow is grouping chemicals into chemical categories.

The main objective of the Toolbox is to allow the user to use (Q)SAR methodologies to group chemicals into categories and to fill data gaps by read-across, trend analysis and (Q)SARs.

The seminal features of the Toolbox are:

- Identification of relevant structural characteristics and potential mechanism or mode of action of a target chemical.
- Identification of other chemicals that have the same structural characteristics and/or mechanism or mode of action.
- Use of existing experimental data to fill the data gap(s).

New features of the QSAR Toolbox version 3.0 are:

- Inclusion of additional data sources
- Advanced search engine
- 22 new mechanistically and endpoint specific profiling schemes
- Quantitative mixtures toxicity prediction
- Tautomeric set prediction
- Prediction accounting for metabolism

- New transformation simulators (autoxidation and hydrolysis)
- Enhanced reporting engine to handle mixtures, tautomers and metabolites

Links

- Home page.
- Download Links
- Tutorials.

GUSAR

Overview

GUSAR software was developed to create QSAR/QSPR models on the basis of the appropriate training sets represented as SDfile contained data about chemical structures and endpoint in quantitative terms.

GUSAR allows creating of QSAR models based on predicted biological activity profiles of chemical compounds. Each chemical compound is represented as a list of MNA descriptors, which are used as input parameters for predicting of the biological activity profiles. PASS algorithm is used to calculate this profile.

QNA descriptors are P and Q values calculated for each atom of molecule. The calculation of P and Q values is based on the connectivity matrix (C) and the standard values of ionization potential (IP) and electron affinity (EA) of atoms in a molecule [1]. The estimation of a target property of chemical compound is calculated as the mean value of the function of P and Q values of the atoms of a molecule in QNA descriptors' space. We have proposed to use two-dimensional Chebyshev polynomials for approximation of the function of P and Q values. So, the independent regression variables are calculated as average values of particular two-dimensional Chebyshev polynomials of P and Q values for molecule atoms.

Links

- Home Page

PreADMET

Overview

PreADMET is a web-based application for predicting ADME data and building drug-like library using in silico method. PreADMET ver 2.0 is also commercially available in the four editions: Descriptors, Endpoint, Standard and Professional.

- Molecular Descriptors Calculation - 1081 diverse molecular descriptors
- Drug-Likeness Prediction - Lipinski'r rule, lead-like rule, Drug DB like rule
- ADME Prediction - caco-2, MDCK, BBB, HIA, plasima protein binding and skin permeability data
- Toxicity Prediction - Ames test and rodent carcinogenicity assay

Links

- Home page

KNIME

Overview

KNIME [naim] is a user-friendly graphical workbench for the entire analysis process: data access, data transformation, initial investigation, powerful predictive analytics, visualisation and reporting. The open integration platform provides over 1000 modules (nodes), including those of the KNIME community and its extensive partner network.

Links

- DownloadLink

Avalon toolkit

Overview

The Avalon Cheminformatics Toolkit contains tools to render and canonicalize SMILES and manipulate MOL file and related formats as well as structure fingerprinting.

Links

- Download page

ChemSpotlight

Overview

ChemSpotlight is a Spotlight metadata importer plugin for Mac OS X, which reads common chemical file formats using the Open Babel chemistry library. Spotlight can then index and search chemical data: molecular weights, formulas, SMILES, InChI, fingerprints, etc.

It's probably easier to show the results from ChemSpotlight than to describe it. ChemSpotlight indexes chemistry files, adds molecular formulas (complete with subscripts and superscripts in the Finder), molecular weight, and a variety of other information for Spotlight searches and "Get Info" windows.

Links

- Download Link

---

goChem

Overview

A library for computational chemistry. goChem is an open-source library for simplifying the common tasks of a computational chemist at a classical and quantum level. Although it is somewhat bio-oriented, it aims to be useful for chemistry in general. goChem provides a nice set of features including geometric analysis, compatibility with several structures and trajectory formats, and interaction with several quantum-chemistry packages.

Links

- Download Link

PowerMV

Overview

A software environment for statistical analysis, molecular viewing, descriptor generation, and similarity search.
Basic Functions:

- Supports MDL SDF format
- Displays molecules in multiple columns.
- Displays properties contained in SD file in a table.
- Anti-alias technology for best picture quality.
- Table of molecule pictures and properties can be exported to Excel (Office XP and above) to generate personalized reports.
- Calculates three types of binary atom pair descriptors and continuous weighed burden numbers.
- Searches over ACL library to determine possible mechanisms or side effects. The user can create and load their personal databases.
- Calculates Drug-like properties like LogP, PSA, MW, HBAs, HBDs, etc.
- Builds regression model using Least Angle Regression (LARS) and LASSO-2
- Builds regression and classification model using Random Forest through graphical interface to R.
- Cluster analysis with KMeans through graphical interface to R.
- Outlier detection using tetrads method (Douglas Hawkins, et al). (Code implemented by Andrew Wong).
- Novel robust single value decomposition (RSVD) for large datasets with missing values or outliers.

Links

- Download Link

Chemspider

Overview

ChemSpider offers a powerful set of web services to access and query the database using your own information systems.

- Search by molecular mass or elemental composition within ChemSpider or within particular data source(s).
- Search by chemical identifier.
- Retrieving information about ChemSpider record(s).
- Retrieving chemical structure thumbnail.
- Generation of a SMILES from a chemical structure
- Generation of a chemical structure from a SMILES
- Generation of an InChI from a chemical structure
- Generation of a chemical structure from an InChI
- Conversion between chemical data formats using OpenBabel

Links

- Home Page

SDF Toolkit

Overview

The purpose of this SDF toolkit is to provide functions to read and parse SDFs, filter, and add/remove properties. It can also read comma separated value (CSV) tables which contain new fields to be added to the SD file. A typical application is to add calculated Log P values or biological data exported from a spreadsheet.

One useful application (at least for us) that has been written with this toolkit: "add_prop_sdf". This script reads an SDF, adds properties from a CSV file and prints out the new SD file.

This toolkit is written in Perl 5.

Links

- Download Link

References and Tutorials

In this section I will list some references and materials that helped me to get started and understand more about cheminformatics in short time. I think it could be useful to help competitors in future contests to get started.

What is a molecular descriptor?

Overview

This is a pdf tutorial that describes in brief what the molecular descriptor is.

Links

- Tutorial Link

Molecular descriptors and chemometrics

Overview

The concept of molecular structure is one the most important concepts in the development of the scientific knowledge of the XX century. As a matter of fact the reasoning based on the molecular structure has been the main engine for the great development of physical chemistry, molecular physics, organic chemistry, quantum chemistry, chemical synthesis, polymer chemistry, medicinal chemistry. By definition, a system is complex when.

Links

- Tutorial Link

Basic requirements for valid molecular descriptors

Overview

Several scientists are involved in searching for new molecular descriptors able to catch new aspects of the molecular structure. This kind of research involves creativity and imagination together with solid theoretical basis to obtain numbers with some structural chemical meaning.

A molecular descriptor can be more or less useful, simple, interpretable, etc., but, in any case, it has to fulfil some mathematical requirements. In particular, the basic properties a molecular descriptor MUST HAVE are.

Links

- Tutorial Link

Chemistry Toolkit Rosetta Wiki

Overview

The Chemistry Toolkit Rosetta is a wiki for sharing how to use different chemistry toolkits for the same set of common tasks. The main focus is on chemical informatics, with toolkits that handle molecular structures, depiction, databases, property analysis, nomenclature, and the like. This includes 3D structure visualization especially as applied to docking and small molecule conformation generation, but the visualization should support bond types. In the future we may add other chemistry tasks, like MMFF energy evaluation, molecular dynamics, spectra analysis, x-ray crystallography, quantum mechanics and more, but don't hold your breath unless you want to help define and evaluate the tasks!

Links

- Wiki Link

Useful and unuseful summaries of regression models

Overview

How to make useful regression model reports and avoid unuseful things: simple definitions of some common regression quantities (coefficient of determination, error standard deviation, F-ratio test in regression, Predictive Error Sum of Squares), examples of questionable regression summaries and proposals of good summaries for regression models.

Links

- Tutorial Link

Defining the Applicability Domain of QSAR models

Overview

QSARs establish a quantitative relationship between chemical structures and their properties. In theory, QSAR models can be used to predict the properties of chemical structures, provided their structural information is available. The rising popularity of QSAR models is also accompanied by a question over their reliable predictions. In theory the applicability of QSAR models to the query chemicals is limited. Reliable predictions are usually confined to those chemicals that are structurally similar to the training compounds used to build the model. The principle of Applicability Domain obliges the users to define the model limitations with respect to its structural domain and response space.

Links

- Tutorial Link