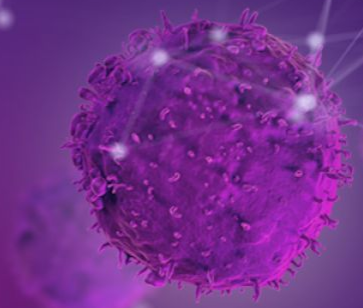
ardigen

Artificial Intelligence & Bioinformatics
for Precision Medicine

group of machine
gmum
learning research

25 November 2019



ardigen

group of machine gmum learning research

Representations of chemical data

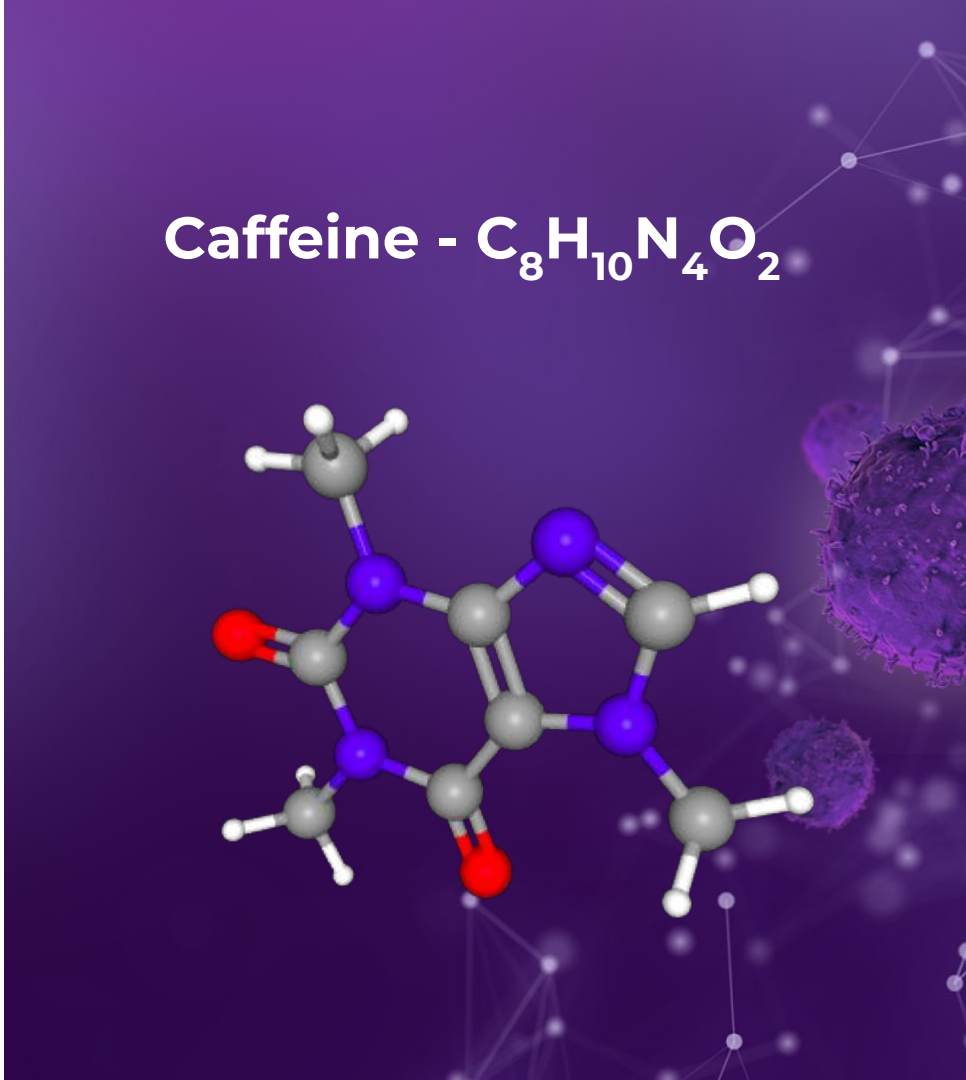
Machine Learning Methods in Cheminformatics

Agenda

- Intro. Why representing chemical data is essential?
- Textual Representation: SMILES
- Numerical representations:
    - MACCS
    - ECFP
    - Spectrophores
- Splitting methods
- How to do it in code? (RDKit)

INTRO

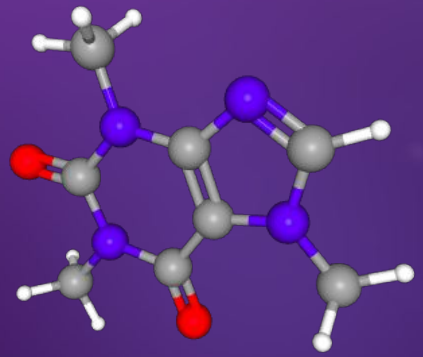
- A typical molecule
- How to feed this data info a ML algorithm?
- Naive molecular representation doesn't include any information about structure
- There is a need for a numerical or at least a textual representation of the data. (Analogical to NLP)



SMILES

SMILES

- Ancient (1980's) representation of molecules
- Preserves structure
- Can be easily fed into RNN or CharCNN model
- Can already be used to classify and even generate molecules (to some extent...)



CN1C=NC2=C1C(=O)N(C(=O)N2C)C

SMILES

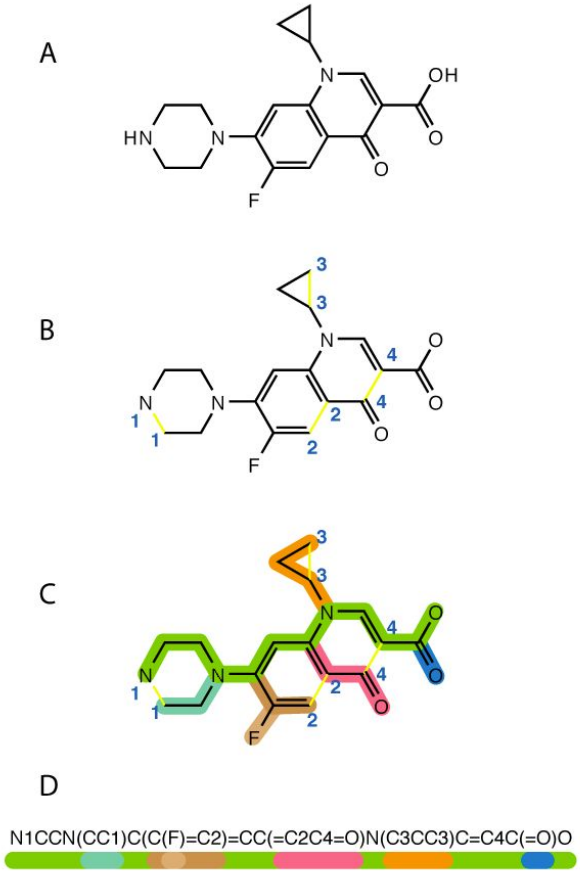


Image:https://pl.wikipedia.org/wiki/SMILES

Caveats

- A model has learn how to distinguish between “S” (sulphur) and “Si” (silicone)
- Doesn’t achieve high accuracy
- Some molecules can have multiple SMILES representations
- It is hard to compare two molecules based only on SMILES

MACCS

MACCS

- Binary string
- Based on a set of questions (keys) about the molecule
- Molecules can be somewhat compared using the Hamming distance between their fingerprints
- The representation is dependant on the selection of keys

- Are there fewer than 3 oxygens?

  - YES -
- Is there a S-S bond?

  - NO -
- Is there a ring of size 5?

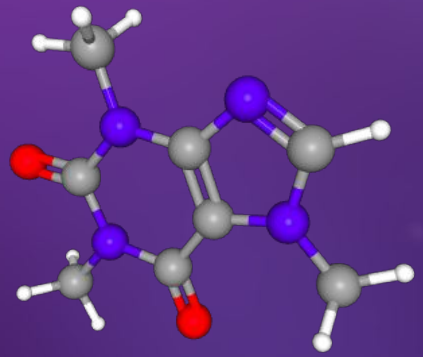
  - YES -
- Is at least one F, Cl, Br, or I present?

  - NO -

RESULT - 1010

MACCS

- Most widely used key dictionary was made by MDL
- It contains 166 keys



00000000000000000000000000000000
00000001100000000000000000000000
00001000000000010101100101000100110 1
11100100011000100100010011100101000 0
00001100011111000111000101011111111 0

ECFP

ECFP

- An iterative algorithm:
  - Assign each atom with an identifier
  - Update each atom's identifiers based on its neighbours
  - Remove duplicates
  - Fold list of identifiers into a bit vector (a Morgan fingerprint)

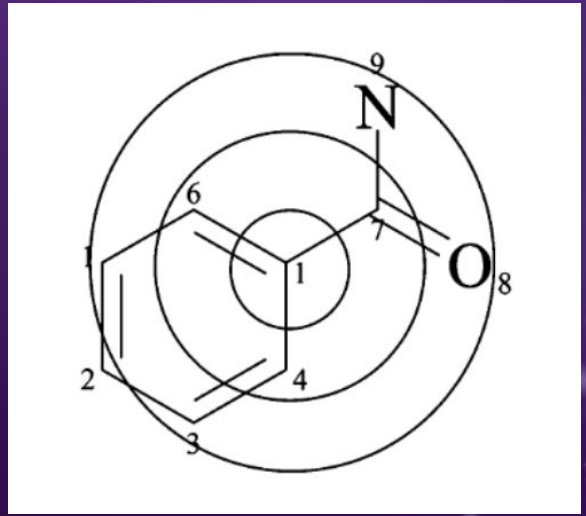


Image: Ridgers, Hahn (2010) Extended-Connectivity Fingerprints

ECFP - Step 1

- Calculate the atom identifier
  - Number of nearest-neighbour non-hydrogen atoms
  - Number of bonds attached to the atom (not including bonds to hydrogens)
  - Atomic number
  - Atomic mass
  - Number of hydrogens connected to the atom
  - Is the atom in a ring (1) or not (0)?

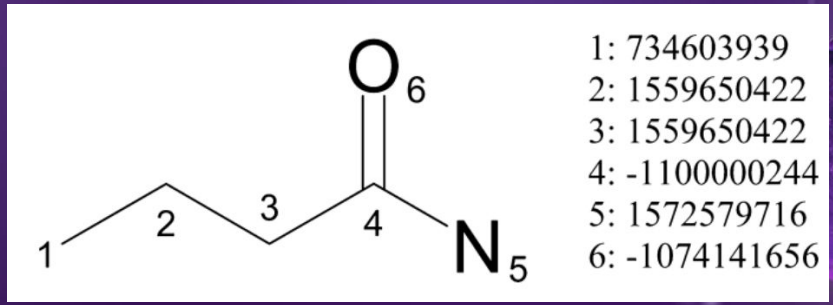


Image: Ridgers, Hahn (2010) Extended-Connectivity Fingerprints

ECFP - Step 2

- At each iteration, for every atom create a list containing its identifier and its neighbours' identifiers
- Append the new identifiers to the old ones
- Remove any duplicates
- Iterate again
- Hash the resulting array to a bit string

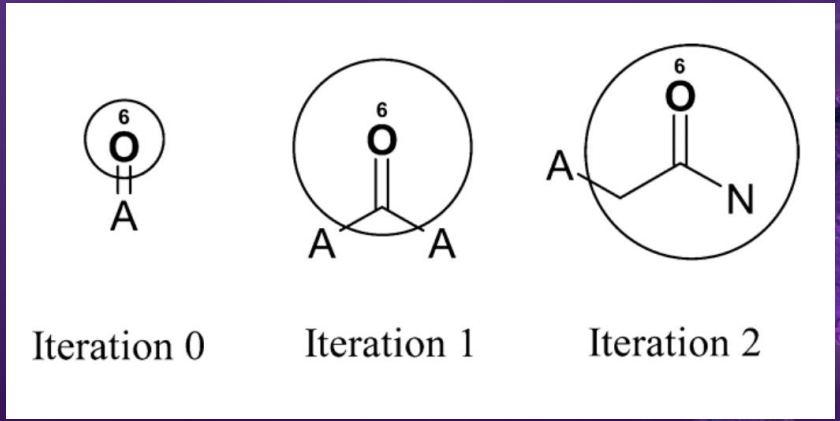


Image: Ridgers, Hahn (2010) Extended-Connectivity Fingerprints

ECFP - an algorithm

Algorithm 1 Circular fingerprints

Input: molecule, radius R, fingerprint length S
Initialize: fingerprint vector $\mathbf{f} \leftarrow \mathbf{0}_S$
for each atom a in molecule
    $\mathbf{r}_a \leftarrow g(a)$ ▷ lookup atom features
for $L = 1$ to R ▷ for each layer
    for each atom a in molecule
        $\mathbf{r}_1 \dots \mathbf{r}_N = \text{neighbors}(a)$
        $\mathbf{v} \leftarrow [\mathbf{r}_a, \mathbf{r}_1, \dots, \mathbf{r}_N]$ ▷ concatenate
        $\mathbf{r}_a \leftarrow \text{hash}(\mathbf{v})$ ▷ hash function
        $i \leftarrow \text{mod}(r_a, S)$ ▷ convert to index
        $\mathbf{f}_i \leftarrow 1$ ▷ Write 1 at index
Return: binary vector $\mathbf{f}$

Duvenaud et al. (2015) Convolutional Networks on Graphs for Learning Molecular Fingerprints

ECFP

- Depending of a particular problem, we can use a different number of iterations.
- If we want to capture entire ring structure, we have to use at least 4 iterations
- For tasks relying on atoms and their bonds, 2 iterations might be enough

SPECTROPHORES

Spectrophores

- One dimensional vector computed from 3D properties
- Typically, a vector of 48 real numbers, describing interaction of the molecule with some environment
- They are popular in ligand-based virtual screening

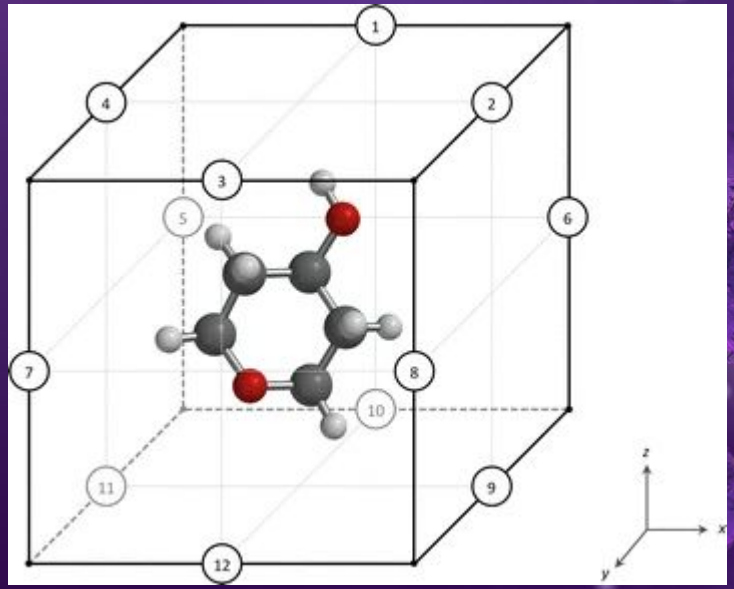


Image: Gladysz et al. (2018) Spectrophores as one-dimensional descriptors calculated from three-dimensional atomic properties: applications ranging from scaffold hopping to multi-target virtual screening

Spectrophores

- Calculated as an interaction properties between molecule and an artificial cage by a molecular conformation
- Properties of atoms used to calculate the fingerprint include:
  - atomic partial charges
  - atomic lipophilicity indices
  - atomic shape deviations
  - atomic softness properties

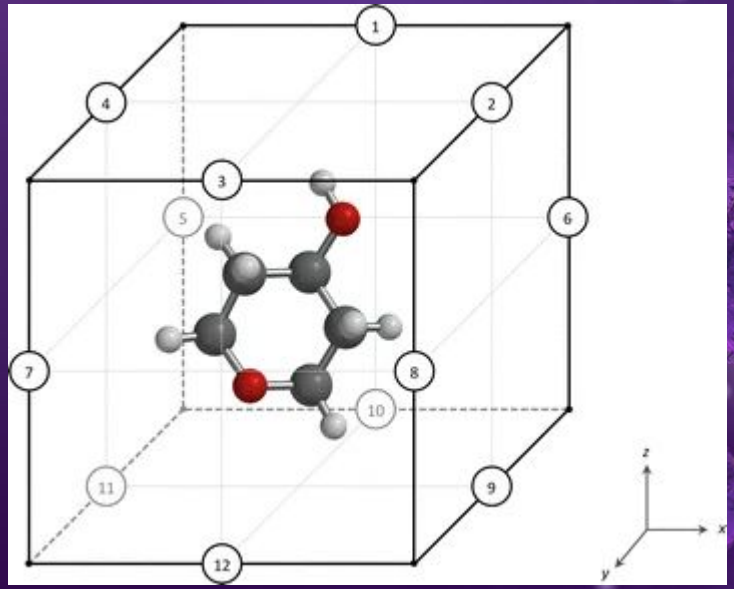


Image: Gladysz et al. (2018) Spectrophores as one-dimensional descriptors calculated from three-dimensional atomic properties: applications ranging from scaffold hopping to multi-target virtual screening

SPLITTING METHODS

Splitting methods

- Using random splitting methods is often not desirable for chemical tasks, since problems usually involve working with novel molecules (e. g. in drug discovery)
- Scaffold splitting divides the dataset into parts with distinct molecules structure based on scaffolds

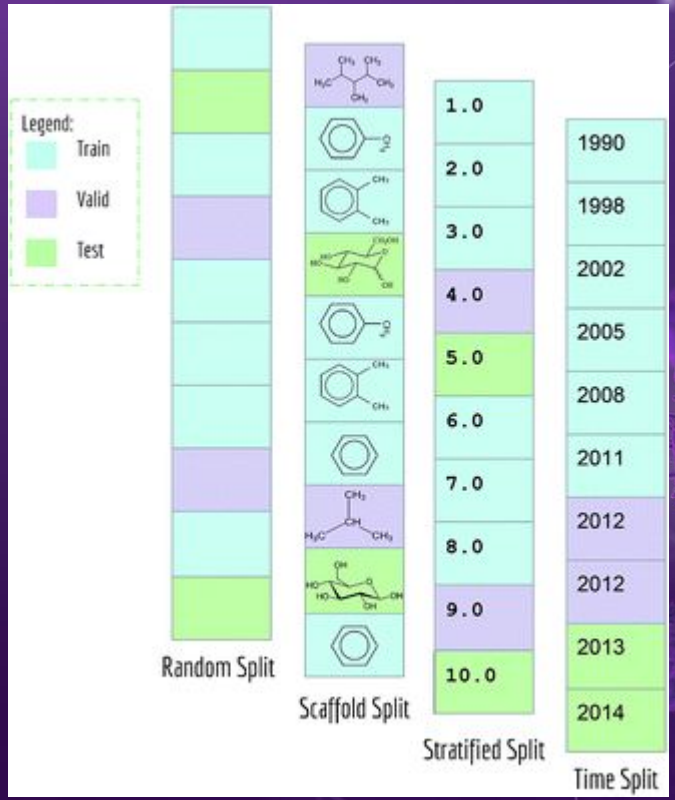
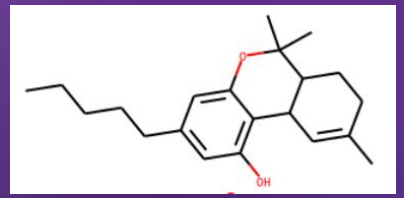
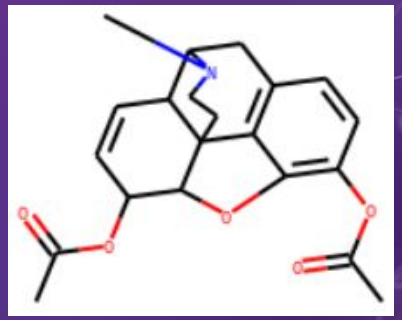


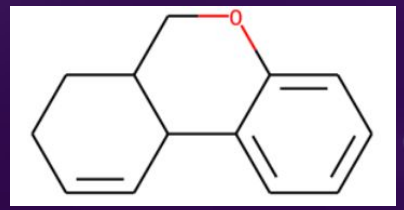
Image: Wu et al. (2017) MoleculeNet: a benchmark for molecular machine learning

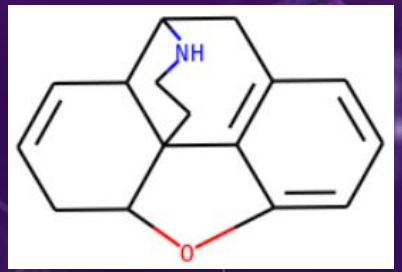
What are scaffolds?

- Scaffolds are a structure based reduced representations for molecules
- The most popular Murcko scaffold is defined as ring systems with connecting atoms









HOW TO DO IT IN CODE?

RDKit

group of machine gmum learning research ardigen

- Open-Source, widely used software for Cheminformatics and Machine Learning
- Contains tools for many cheminformatics tasks, like SMILES, fingerprinting, pharmacophore representation, molecule and atom properties etc.
- API in C++ and Python

Open Babel

- Toolbox created to work with chemical data from different sources
- Its main use is to convert between different formats of storing chemical data
- Can be used as standalone tool or programming language library

1. Install RDKit

group of machine gmum learning research ardigen

conda install -c conda-forge **rdkit**

1. Install Open Babel

group of machine gmum learning research ardigen

conda install -c openbabel **openbabel**

RDKit Basics

- RDKit representation of molecule can be created from SMILES or .mol file:

```
>>> from rdkit import Chem

>>> m = Chem.MolFromSmiles('Cc1ccccc1')

>>> m = Chem.MolFromMolFile('data/input.mol')
```

RDKit - MACCS

- Fingerprint creation operated directly on RDKit representation:

```
>>> from rdkit.Chem import MACCSkeys
>>> m = Chem.MolFromSmiles('Cc1ccccc1')
>>> maccs = MACCSkeys.GenMACCSKeys(m)
>>> print(maccs.ToBitString())
10101011101111011...
```

RDKit - ECFP

- The same for ECFP:

```
>>> from rdkit.Chem import AllChem
>>> m = Chem.MolFromSmiles('Cc1ccccc1')
>>> fp = AllChem.GetMorganFingerprint(m1, 2)
>>> fp1 = AllChem.GetMorganFingerprintAsBitVect(m1, 2, nBits=1024)
```

Open Babel - Spectrophores

- Can be used as a tool from command line:

  >>> **obspectrophore** -i input.smi

Open Babel - Spectrophores

- Or using python bindings:

```
>>> import openbabel
>>> sp = openbabel.OBSpectrophore()
>>> conv = openbabel.OBConversion()
>>> mol = openbabel.OBMol()
>>> conv.ReadFile(mol, 'input.sdf')
>>> print(sp.GetSpectrophore(mol))
(1.756782876340235, 1.5915547901054223, 1.5932670837515774, 3.6518752424951533, ... )
```

https://tinyurl.com/vkoxwcv

ardigen

Artificial Intelligence & Bioinformatics
for Precision Medicine

group of machine
gmum
learning research

25 November 2019