Setting up

Linux

```
conda create -n neovarsity python=3.7 -y
conda activate neovarsity
conda config --add channels conda-forge
conda install -c rdkit rdkit
conda install -c sepandhaghighi pycm -y
conda install -c conda-forge imbalanced-learn -y
conda install molvs -y
conda install -c conda-forge padel
```

Please note that throughout the curriculum, we will also install some additional packages as we need them. However, for the time being, installing these packages will suffice.

> **Pro tip:** You can copy and paste all the commands at once and then press enter. In the end of the command e.g. `conda create -n neovarsity python=3.7 -y`, the **"-y"** option is a flag that stands for "**yes**". It is used to automatically answer "yes" to any prompts or confirmations during the execution of the command where your interaction is not desired.

Descriptions of the Installation Steps

First, we will create a new environment called 'neovarsity' using the command `conda create -n neovarsity python=3.7 -y`. This will create a new environment called `neovarsity` with Python 3.7 installed.

> Please note that we have utilized Python version 3.7, which is not the latest version of Python available. The reason for this choice is to ensure compatibility with various packages that may not be compatible with newer versions of Python.

Next, we will activate this environment using the command `conda activate neovarsity`. This will ensure that any packages we install will be installed in the `neovarsity` environment rather than the default environment.

To ensure that we have access to all the required packages, we will add the "conda-forge" channel using the command `conda config --add channels conda-forge`.

We will then install the RDKit package, which is a collection of cheminformatics and machine learning tools, using the command `conda install -c rdkit rdkit -y`.

RDKit is widely used in the field of chemistry and drug discovery for tasks such as molecule rendering, substructure searching, and similarity searching. With this command, you will be able to use RDKit in your Python projects for cheminformatics and machine learning tasks.

To evaluate machine learning models, we will install the pycm package, which provides tools for confusion matrix analysis, using the command `conda install -c sepandhaghighi pycm -y`.

Finally, we will install the imbalanced-learn package using the command `conda install -c conda-forge imbalanced-learn -y`. This package provides tools for dealing with imbalanced datasets, which are common in many real-world applications.

We will also install the molvs package using the command `conda install molvs -y`, which provides tools for molecule standardization, including tautomer enumeration and stereochemistry handling.

The command `conda install -c conda-forge padel` installs the Padel Descriptor package, which is a collection of public-domain software programs for calculating molecular descriptors. The package is installed from the `conda-forge` channel on Anaconda, specified with the `-c conda-forge` option.

The Padel Descriptor package can be used for various cheminformatics tasks, such as compound classification, virtual screening, and similarity searching. It calculates a range of molecular descriptors, which are quantitative descriptions of molecular properties that can be used to predict various properties or activities of compounds.

Please keep in mind that the installation of Jupyter Notebook, Sci-kit Learn and Matplotlib is also necessary. Ideally, these packages are already pre-installed in your Anaconda environment. You can quickly check that by executing the `conda list` command in your Linux/Ubuntu terminal. However, if these packages are not present, you can install them using the following commands.

We can install the Jupyter Notebook using the command `conda install jupyter notebook -y`. This is a popular tool used for interactive computing, and it allows you to create and share documents that contain live code, equations, visualizations, and narrative text.

We can install scikit-learn, a popular machine learning library, using the command `conda install scikit-learn -y`. This library provides tools for data mining and data analysis, and is widely used in the field of machine learning.

We can install the matplotlib package, which is a plotting library used for data visualization, using the command `conda install matplotlib -y`.

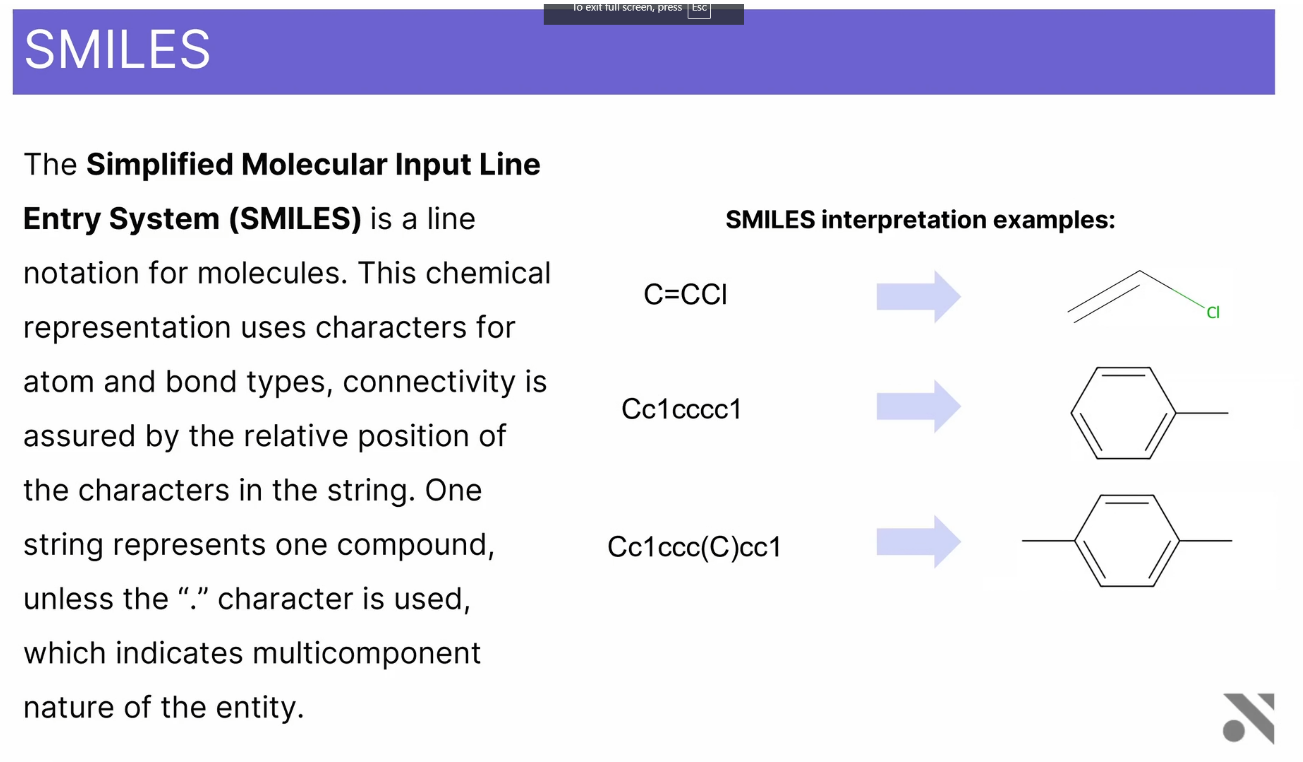
And that's it! By following these installation steps, you should now have a fully functional environment for data science and machine learning suited for Cheminformatics and Machine Learning for Drug Discovery, with all the necessary packages installed.

Throughout the curriculum, we will also install some additional packages as we need them.

> Encountering installation issues? Reach out to your technical support team or seek assistance from the Slack Community!

File formats

SMILES



SMILES

The **Simplified Molecular Input Line Entry System (SMILES)** is a line notation for molecules. This chemical representation uses characters for atom and bond types, connectivity is assured by the relative position of the characters in the string. One string represents one compound, unless the "." character is used, which indicates multicomponent nature of the entity.

SMILES interpretation examples:

C=CCl

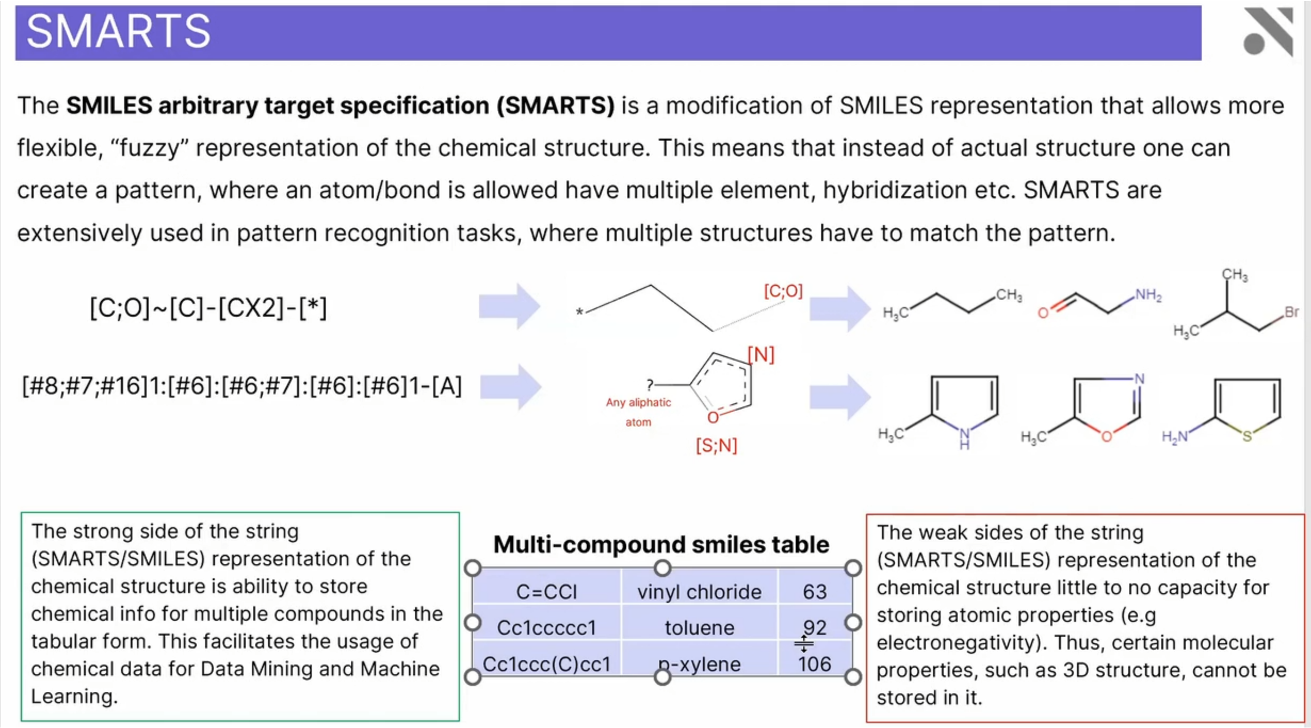
Cc1cccc1

Cc1ccc(C)cc1

other points

- Cannot keep track of atom electronegativity and atom specific information
- some softwares like chemaxon requires SMILES to be in first column. take note of whether need headers

SMARTS



SMARTS

The **SMILES arbitrary target specification (SMARTS)** is a modification of SMILES representation that allows more flexible, “fuzzy” representation of the chemical structure. This means that instead of actual structure one can create a pattern, where an atom/bond is allowed have multiple element, hybridization etc. SMARTS are extensively used in pattern recognition tasks, where multiple structures have to match the pattern.

[C;O]~[C]-[CX2]-[*]

[#8;#7;#16]1:[#6]:[#6;#7]:[#6]:[#6]1-[A]

The strong side of the string (SMARTS/SMILES) representation of the chemical structure is ability to store chemical info for multiple compounds in the tabular form. This facilitates the usage of chemical data for Data Mining and Machine Learning.

Multi-compound smiles table

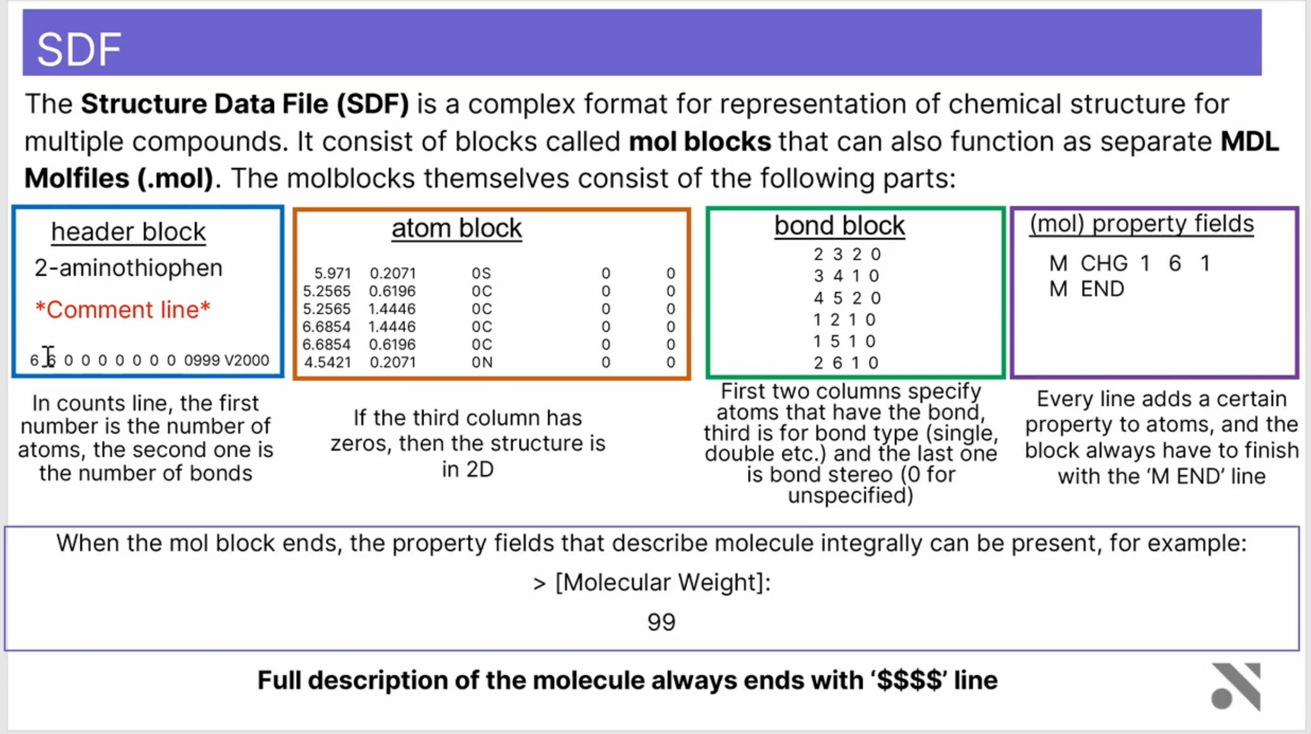
C=CCl	vinyl chloride	63
Cc1ccccc1	toluene	92
Cc1ccc(C)cc1	p-xylene	106

The weak sides of the string (SMARTS/SMILES) representation of the chemical structure little to no capacity for storing atomic properties (e.g electronegativity). Thus, certain molecular properties, such as 3D structure, cannot be stored in it.

other notes

- fuzzy representation

SDF



SDF

The **Structure Data File (SDF)** is a complex format for representation of chemical structure for multiple compounds. It consist of blocks called **mol blocks** that can also function as separate **MDL Molfiles (.mol)**. The molblocks themselves consist of the following parts:

header block

```
2-aminothiophen
*Comment line*
6 6 0 0 0 0 0 0 0 0999 V2000
```

In counts line, the first number is the number of atoms, the second one is the number of bonds

atom block

```
  5.971   0.2071     0S        0      0
5.2565    0.6196     0C        0      0
5.2565    1.4446     0C        0      0
6.6854    1.4446     0C        0      0
6.6854    0.6196     0C        0      0
4.5421    0.2071     0N        0      0
```

If the third column has zeros, then the structure is in 2D

bond block

```
2 3 2 0
3 4 1 0
4 5 2 0
1 2 1 0
1 5 1 0
2 6 1 0
```

First two columns specify atoms that have the bond, third is for bond type (single, double etc.) and the last one is bond stereo (0 for unspecified)

(mol) property fields

```
M  CHG  1   6   1
M  END
```

Every line adds a certain property to atoms, and the block always have to finish with the ‘M END’ line

When the mol block ends, the property fields that describe molecule integrally can be present, for example:

> [Molecular Weight]:

99

Full description of the molecule always ends with ‘$$$$’ line

PDB

other notes

Standardization

- rdkit + knime
- https://github.com/rdkit/rdkit/blob/master/Docs/Notebooks/MolStandardize.ipynb

Chemical search and filter

- Datawarrior
- rdkit + SMARTS

Descriptors

Max common substructures

MinMax Diversity - train test split

Chemical Similarity

Activity Cliffs

Molecular graphs + Visualization

Clustering Techniques

QSAR

Chemical Databases

Chemical Ring Mining Databases

Chemical Structure Transformations

Building Combinatorial Database

Bioisoster Shape Similarity Analysis

Virtual Screening

Capstone