flamingo Documentation

Release 0.3.1

Felipe Zapata

Sep 11, 2023

Contents:

CHAPTER 1

flamingo

Compute and filter molecular properties. See documentation.

1.1 Installation

- Download miniconda for python3: miniconda.
- Install according to: installConda.
- Create a new virtual environment using the following commands:
  - `conda create -n flamingo`
- Activate the new virtual environment
  - `conda activate flamingo`

To exit the virtual environment type `conda deactivate`.

1.1.1 Dependencies installation

- Type in your terminal:

```
conda activate flamingo
```

Using the conda environment the following packages should be installed:

- install RDKit and H5PY:
  - *conda install -y -q -c conda-forge h5py rdkit*

1.1.2 Package installation

Finally install the package:

- Install **flamingo** using pip: - `pip install nlesc-flamingo`

Now you are ready to use *flamingo*.

Contributing

If you want to contribute to the development of flamingo, have a look at the contribution guidelines.

License

Credits

This package was created with Cookiecutter and the NLeSC/python-template.

CHAPTER 2

Tutorial in Silico Screening

This tutorial covers how to perform insilico filtering of a set of molecules represented as smiles. Table *smiles to filter* contains some smiles examples representing molecules with different functional groups.

Table 1: smiles to filter

smiles
CN1C=NC2=C1C(=O)N(C(=O)N2C)C
OC(=O)C1CNC2C3C4CC2C1N34
C1=CC=CC=C1
OC(=O)C1CNC2COC1(C2)C#C
CCO
CCCCCCCCC=CCCCCCCCC(=O)O
CC(=O)O
O=C(O)Cc1ccccc1
CC(C(=O)O)O

The filtering process consists in excluding (or including) a set of molecules based on structural characteristics like their functional groups or derived properties like bulkiness.

To run the simulation, the user must provide two files: one containing the smiles that she wants to filter and another file containing the values of the properties used as filters.

2.1 Simulation input

The smiles input should be in csv format like

```
,smiles
,CC(=O)O
,CCC(=O)O
```

The properties specification file to perform the filtering must be a yaml file following the subsequent schema yaml:

```
smiles_file:
  smiles.csv

core:
  "Cd68Se55.xyz"

anchor:
  "O(C=O)[H]"

batch_size: 1000

filters:
  include_functional_groups:
    groups:
       - "[CX3](=O)[OX2H1]" # Include carboxylic acids
    maximum: 1
  exclude_functional_groups:
    groups:
       - "[NX3]"  # Exclude tertiary amines
       - "C#C"    # Exclude triplet Carbon-Carbon bonds
  scscore:
    lower_than:
      3.0
  bulkiness:
    lower_than: 20
```

The *smiles_file* entry contains the path to the files containing the smiles. The other keywords will be explain in the following sections.

2.2 Available filters

Note: The filters are run in sequential order, meaning that second filter is applied to the set of molecules remaining after applying the first filters, the third filter is applied after the second and so on.

2.2.1 1. Include and exclude function groups

The *include_functional_groups* and *exclude_functional_groups* as their names suggest keep and drop molecules based on a list of functional groups represented as SMARTS.

the *maximum* keyword indicates what is the maximum number of functional groups that can be present.

2.2.2 2. Synthesizability scores

The scscore is a measure of synthetic complexity. It is scaled from 1 to 5 to facilited human interpretation. See the scscore paper for further details.

2.2.3 3. Bulkiness

Assuming that a given molecule can be attached to a given surface, the bulkiness descriptor gives a measure of the volumen occupied by the molecule from the anchoring point extending outwards as a cone. It requires the *core*

keywords specifying the surface to attach the molecule and the *anchor* functional group used as attachment. See the the CAT bulkiness for further information.

2.3 Running the filtering script

To perform the screening you just need to execute the following command :: smiles_screener -i path_to_yaml_input.yml

2.4 Job distributions and results

For a given filter, **Flamingo** will try to compute the molecular properties in parallel since properties can be computed independently for each molecule. Therefore **Flamingo** split the molecular set into batches that can be computed in parallel. The *batch_size* keyword is used to control the size of these batches.

After the computation has finished the filtered molecules are stored in the **results** folder in the *current work directory*. In that folder you can find a *candidates.csv* file for each batch containing the final molecules.

CHAPTER 3

Molecular features

CHAPTER 4

Smiles Screener

CHAPTER 5

CAT interface

CHAPTER 6

Indices and tables

- genindex
- modindex
- search