Instructions for the SubNetX package (Network extraction part)

Installation

Requirements

- python 3
- rdkit environment
- networkx

rdkit is required for balance calculation and pattern matching. In case the rdkit is not possible to install, only pathways towards a selected precursor or all model metabolites can be generated.

Please, install rdkit and use rdkit environment for running SubNetX. First install anaconda: https://docs.anaconda.com/anaconda/install/index.html

Then install rdkit as described here: https://www.rdkit.org/docs/Install.html.

```
$ conda create -c conda-forge -n my-rdkit-env rdkit $ conda activate my-rdkit-env
```

Since networkx package is not part of the default rdkit environment, install it to the environment as follows when the environment is activated:

```
$ conda install networkx
```

Download repository

```
$ git clone https://c4science.ch/source/biosubnet.git
$ git clone https://github.com/EPFL-LCSB/XXX.git - for github release
```

If you are installing on macOS, make sure you have Homebrew installed, otherwise you might get "git: 'lfs' is not a git command." error. Once you installed Homebrew, run

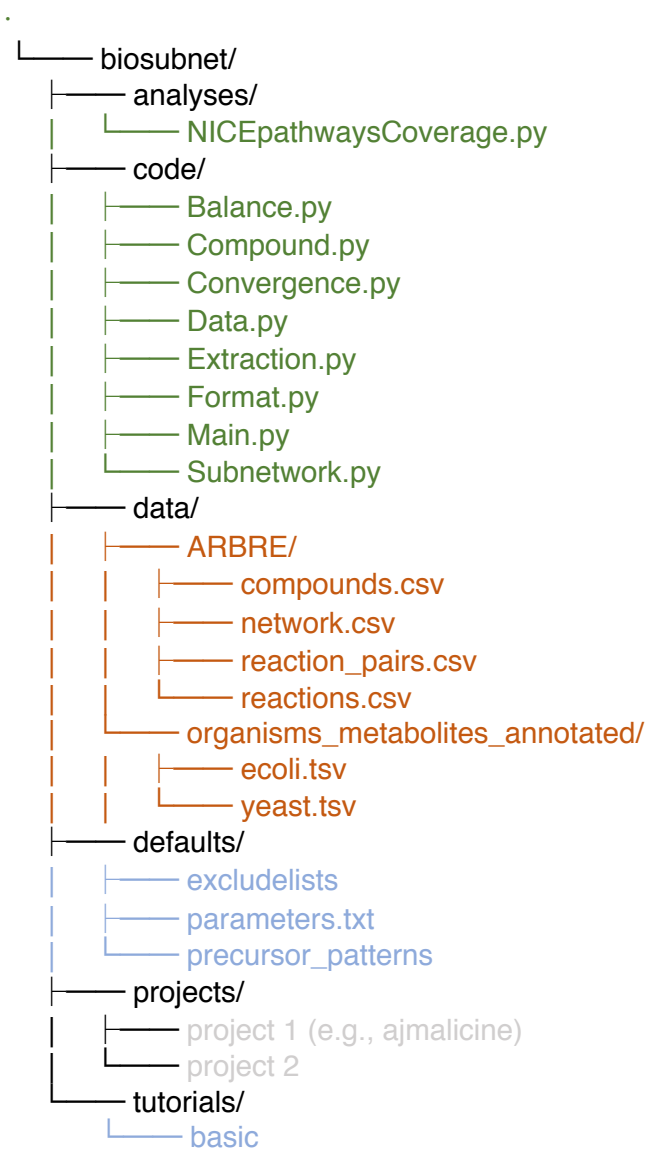
```
$ brew install git-lfs $ run git-lfs install
```

Note

Data files are stored using git large file storage (lfs). The make file will install git lfs automatically. However, if lfs was not installed previously, the repository has to be updated after installation. This is needed to retrieve the data files from the repository after installation.

```
$ git-lfs pull
```

Folder structure



- Analyses folder includes the scripts used for analyses of the SubNetX output in the article
- Data folder includes the reaction network data (in this case data is copied from ARBRE) and organisms_metabolites_annotated including mapping to the reaction network identifiers.
- Defaults folder includes the default parameters (parameters that will be used if no specific parameters provided)
- Tutorials folder includes the parameters used in the case studies of the article

Usage

- create a folder with the name of your project in the "projects" directory (e.g. .../biosubnet/projects/ajmalicine)
- copy the parameters.txt file from the .../biosubnet/defaults folder to your project folder

```
.
└── project 1/
    └── parameters.txt
```

- follow the instructions for the parameters.txt adjustment specified in the parameters.txt file (for more details consult the manuscript)

Execution

Run the code as following

```
$ cd code
$ python3 Main.py {projectname}
```

e.g.,

A. Searching for the basic simplest subnetwork for the set of selected compounds:

A.1. Copy the folders from tutorials/basic to projects/ folder

```
$ cp ../tutorials/basic/* projects/
```

A.2. Execute the script as following:

```
$ python3 Main.py ajmalicine
$ python3 Main.py benzoyl_cinnamate
$ python3 Main.py benzoylbenzoate
$ python3 Main.py berberine
$ python3 Main.py N_cinnamoyl_serotonin
$ python3 Main.py Quercetin_3_O_6_acetylglucoside
$ python3 Main.py scopolamine
$ python3 Main.py strictosidine
```

You will get the following output for each of the projects:

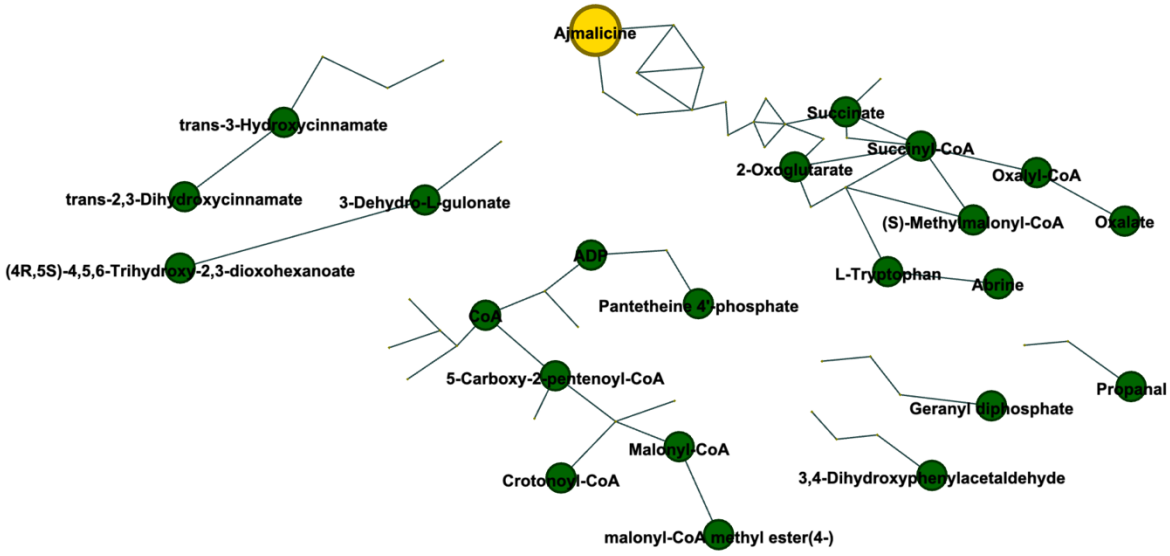
```
.
└── project 1/
    ├── auxilary_output/
    │   ├── boundaries/
    │   │   ├── boundary_no_path.txt
    │   │   ├── branching_points_1_comp.txt
    │   │   ├── branching_points_2plus_comp.txt
```

```
│   │   ├── round1.txt
│   │   └── ...
│   ├── NICEpathways_linear.txt
│   ├── reaction_ids.txt
│   ├── subnetworkConverged.gpickle
│   └── subnetworkExtracted.gpickle
├── figures/
│   ├── convergedIslands_nodesNum.png
│   ├── graph_file_for_gephi.gdf
│   └── graph_file_for_gephi.tsv
├── output_optimization_input/
│   ├── compounds.tsv
│   ├── molfiles/
│   │   ├── cmpId.mol
│   │   └── ...
│   └── reactions.tsv
├── parameters.txt
└── stats/
    ├── converged_graph_stats.csv
    ├── covergeance_rounds_stats.csv
    └── subnetwork_extraction_stats.csv
```

- Main output - output_optimization_input folder: should be copied as an input for the optimization stage
- parameters.txt – parameters.txt file that was copied and adjusted from default
- stats – folder containing number of edges, compounds, execution time for every stage
- auxiliary_output folder contains intermediate output. If no analysis of the intermediate stages results required, one can delete this folder.
- figures – contains output for visual analysis of the extracted subnetwork.

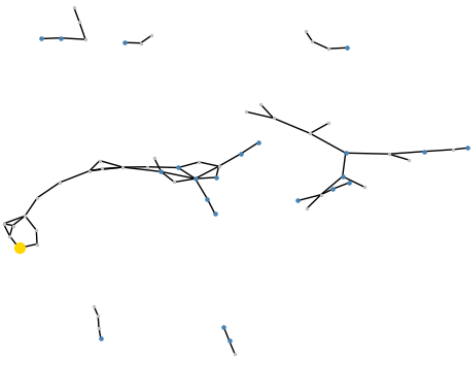
The generation is represented by the color and is labeled in the edges part of the .gdf file in "color VARCHAR" column. You can install Gephi from https://gephi.org. Example of the automatic networkx vs Gephi manually adjusted visualization of the network extraction results are presented below. In both cases, the target compound is in gold color, the organism metabolites are blue/green, the intermediate metabolites are gray.

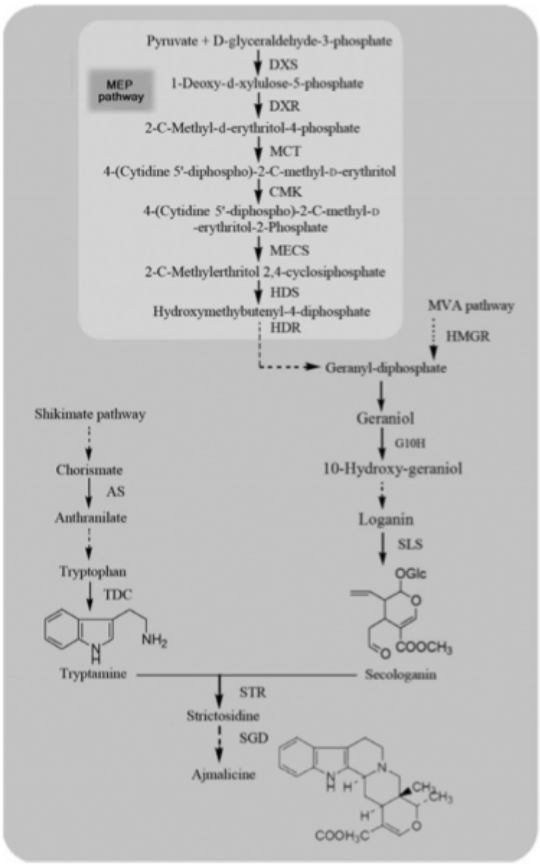
Gephi output:



- Automatic output:

* Characterized PW (PMID: 24237015)





Finishing work with SubNetX

Deactivate your rdkit environment as follows:

```
$ conda deactivate
```

Data

The data files used are taken from ARBRE:
Publication: https://doi.org/10.1016/j.ymben.2022.03.013
Git: https://github.com/EPFL-LCSB/ARBRE

Any data in the same format can be used for the SubNetX. In case alternative dataset is required, the data can be compiled from known reaction databases or from computationally predicted reaction set, as long as information for balance calculation is provided.