8th Advanced In Silico Drug Design workshop

27 - 31 January 2025
Olomouc, Czech Republic

Univerzita Palackého v Olomouci

EasyDock

Pavel Polishchuk, Guzel Minibaeva

Institute of Molecular and Translational Medicine
Faculty of Medicine and Dentistry
Palacky University

pavlo.polishchuk@upol.cz
qsar4u.com

Minibaeva *et al. Journal of Cheminformatics* (2023) 15:102
https://doi.org/10.1186/s13321-023-00772-2

Journal of Cheminformatics

SOFTWARE Open Access

EasyDock: customizable and scalable docking tool

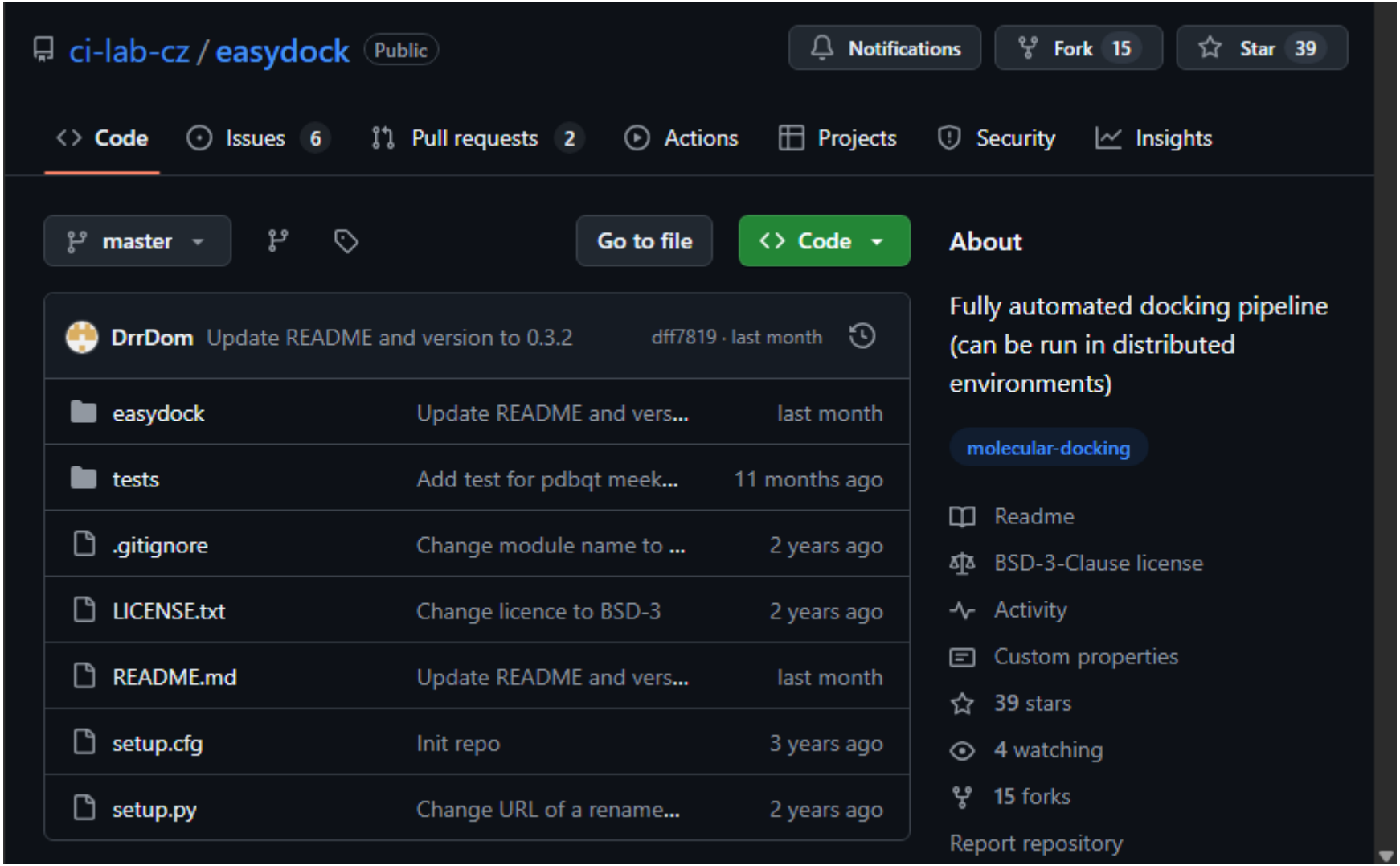
Guzel Minibaeva[1], Aleksandra Ivanova[1] and Pavel Polishchuk[1*]

Abstract

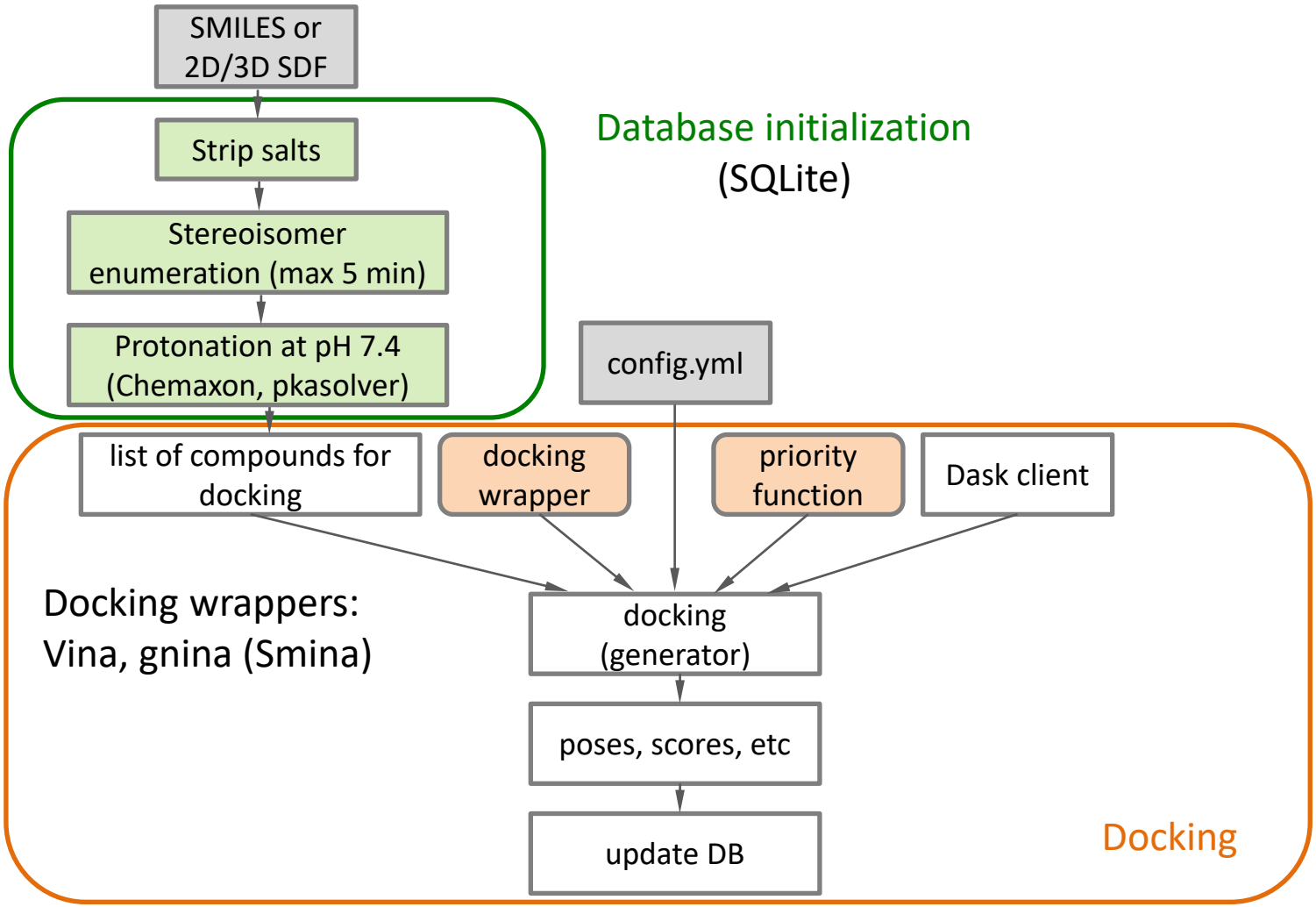
Docking of large compound collections becomes an important procedure to discover new chemical entities. Screening of large sets of compounds may also occur in de novo design projects guided by molecular docking. To facilitate these processes, there is a need for automated tools capable of efficiently docking a large number of molecules using multiple computational nodes within a reasonable timeframe. These tools should also allow for easy integration of new docking programs and provide a user-friendly program interface to support the development of further approaches utilizing docking as a foundation. Currently available tools have certain limitations, such as lacking a convenient program interface or lacking support for distributed computations. In response to these limitations, we have developed a module called EasyDock. It can be deployed over a network of computational nodes using the Dask library, without requiring a specific cluster scheduler. Furthermore, we have proposed and implemented a simple model that predicts the runtime of docking experiments and applied it to minimize overall docking time. The current version of EasyDock supports popular docking programs, namely Autodock Vina, gnina, and smina. Additionally, we implemented a supplementary feature to enable docking of boron-containing compounds, which are not inherently supported by Vina and smina, and demonstrated its applicability on a set of 55 PDB protein-ligand complexes.

Keywords High-throughput molecular docking, Distributed docking, Boron-containing compound docking, AutoDock Vina, Gnina

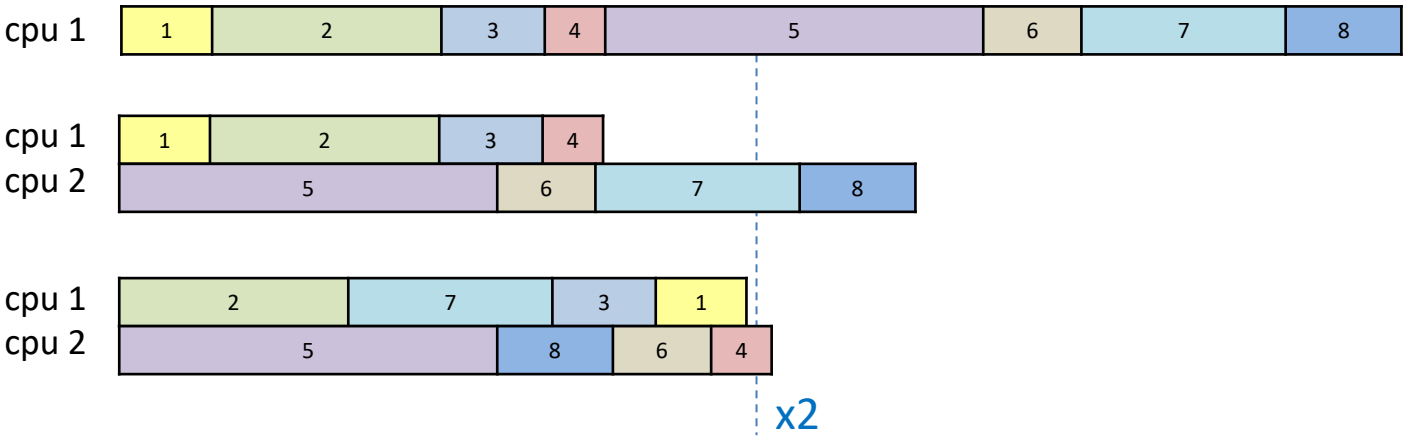
https://github.com/ci-lab-cz/easydock



https://github.com/ci-lab-cz/easydock@dev



Minimization of docking run time



$$\text{time (s)} = 465.979 - 59.714 \times \text{RTB} - 0.375 \times \text{RTB}^2 - 36.723 \times \text{HAC} + 0.745 \times \text{HAC}^2 + 3.48 \times \text{RTB} \times \text{HAC}$$

RTB – the number of rotatable bonds
HAC – the number of heavy atoms

$R^2_{test} = 0.926$ (Vina)

config.yml

Vina

```
protein: /path/to/protein.pdbqt
protein_setup: /path/to/grid.txt
exhaustiveness: 8
seed: 0
n_poses: 5
ncpu: 5
```

Gnina

```
script_file: /path/to/gnina_executable
protein: /path/to/protein.pdbqt
protein_setup: /path/to/grid.txt
exhaustiveness: 8
scoring: default
cnn_scoring: rescore
cnn: dense_ensemble
n_poses: 10
addH: False
ncpu: 1
seed: 0
```

Run docking

Initialization of a database

```
run_dock -i input.smi -o output.db -c 4 --protonation pkasolver
```

Docking of an initialized database

```
run_dock -o output.db --program vina --config config.yml -c 4 --sdf
```

Initialize and docking together

```
run_dock -i input.smi -o output.db -c 4 --protonation pkasolver
--program vina --config config.yml --sdf
```

Features

- the major script run_dock supports docking with vina and gnina (gnina also supports smina and its custom scoring functions)
- can be used as a command line utility or imported as a python module
- input molecules are checked for salts and attempted to fix by SaltRemover
- stereoisomers can be enumerated for unspecified chiral centers and double bonds
- several protonation options: chemaxon and pkasolver
- supports distributed computing using dask library
- supports docking of boron-containing compounds using vina and smina (boron is replaced with carbon before docking and returned back)
- all outputs are stored in an SQLite database
- interrupted calculations can be continued by invoking the same command or by supplying just a single argument - the existing output database
- get_sdf_from_dock_db is used to extract data from output DB

Tutorial

```
run_dock -i cdk2.smi -o 1.db --protonation pkasolver -c 1 --program vina
--config vina_config.yml
```

vina_config.yml

```
protein: 2btr.pdbqt
protein_setup: cdk2_box.txt
exhaustiveness: 4
seed: 120
n_poses: 1
ncpu: 8
```