



Teach Open CADD Update

RDKit UGM 2021

Dominique Sydow

Volkamer Lab

Charité, Berlin

www.volkamerlab.org

TeachOpenCADD in a nutshell

What? Pipelines for common CADD* tasks using open resources

How? Jupyter Notebooks (Python) & KNIME workflows

For what? Teaching, learning & starting point for research projects

Who? Beginners & advanced users

*CADD = computer-aided drug design

TeachOpenCADD links


Jupyter Notebooks

-  <https://github.com/volkamerlab/teachopencadd>
-  <https://projects.volkamerlab.org/teachopencadd/>
-  [Sydow et al. *J. Chem.* \(2019\)](#)

KNIME Workflows

-  <https://hub.knime.com/volkamerlab/spaces/Public/latest/TeachOpenCADD>
-  [Sydow and Wichmann et al. *JCIM* \(2019\)](#)

Teaching with TeachOpenCADD

-  [Sydow, Rodríguez-Guerra, Volkamer. *ACS Symposium Series* \(2021\)](#)

Software | [Open Access](#) | [Published: 08 April 2019](#)

TeachOpenCADD: a teaching platform for computer-aided drug design using open source packages and data


[Dominique Sydow](#), [Andrea Morger](#), [Maximilian Driller](#) & [Andrea Volkamer](#)


Journal of Cheminformatics **11**, Article number: 29 (2019) | [Cite this article](#)
13k Accesses | 5 Citations | 41 Altmetric | [Metrics](#)

JCIM JOURNAL OF
CHEMICAL INFORMATION
AND MODELING

TeachOpenCADD-KNIME: A Teaching Platform for Computer-Aided Drug Design Using KNIME Workflows

Dominique Sydow, Michele Wichmann, Jaime Rodríguez-Guerra, Daria Goldmann, Gregory Landrum, and Andrea Volkamer*

 **Cite this:** *J. Chem. Inf. Model.* 2019, 59, 10, 4083–4086
Publication Date: October 15, 2019
<https://doi.org/10.1021/acs.jcim.9b00662>

Article Views	Altmetric	Citations
4389	14	6

[LEARN ABOUT THESE METRICS](#)

ACS SYMPOSIUM SERIES

Teaching Computer-Aided Drug Design Using TeachOpenCADD

Dominique Sydow, Jaime Rodríguez-Guerra, and Andrea Volkamer*

DOI: 10.1021/bk-2021-1387.ch010
Publication Date: July 27, 2021
[RIGHTS & PERMISSIONS](#)  Subscribed

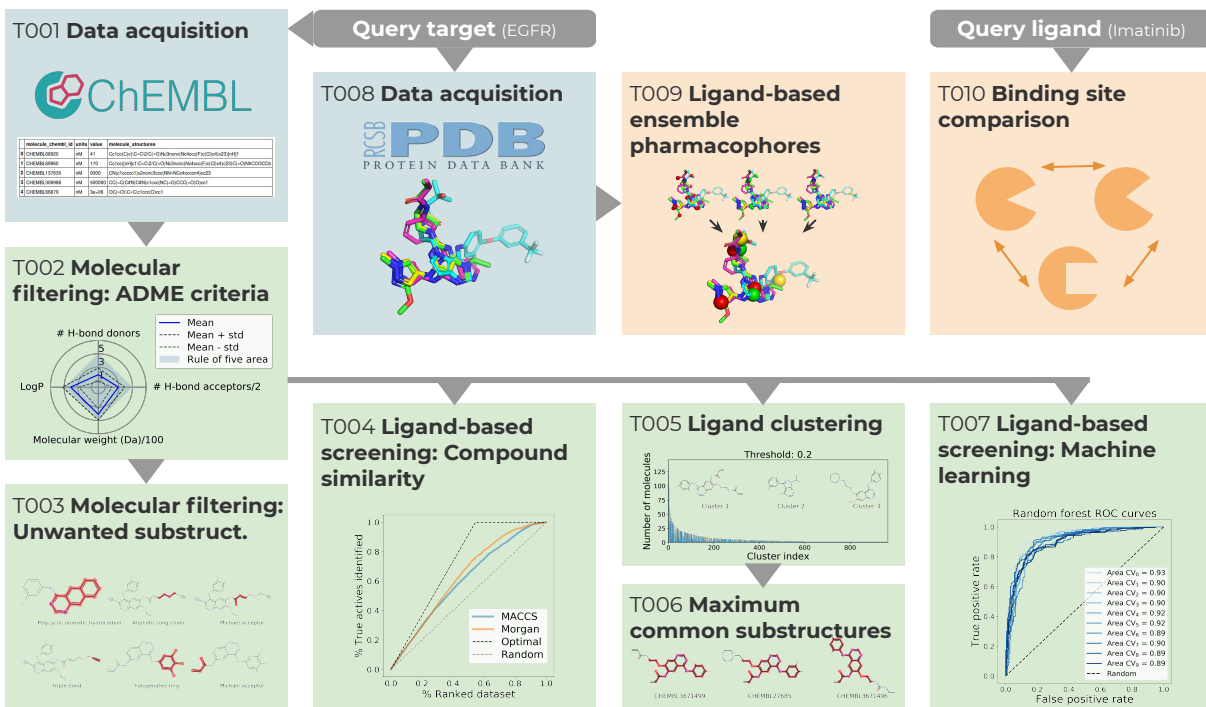
Teaching Programming across the Chemistry Curriculum
Chapter 10, pp 135-158

Chapter Views	Citations
25	-

[LEARN ABOUT THESE METRICS](#)

TeachOpenCADD 2019

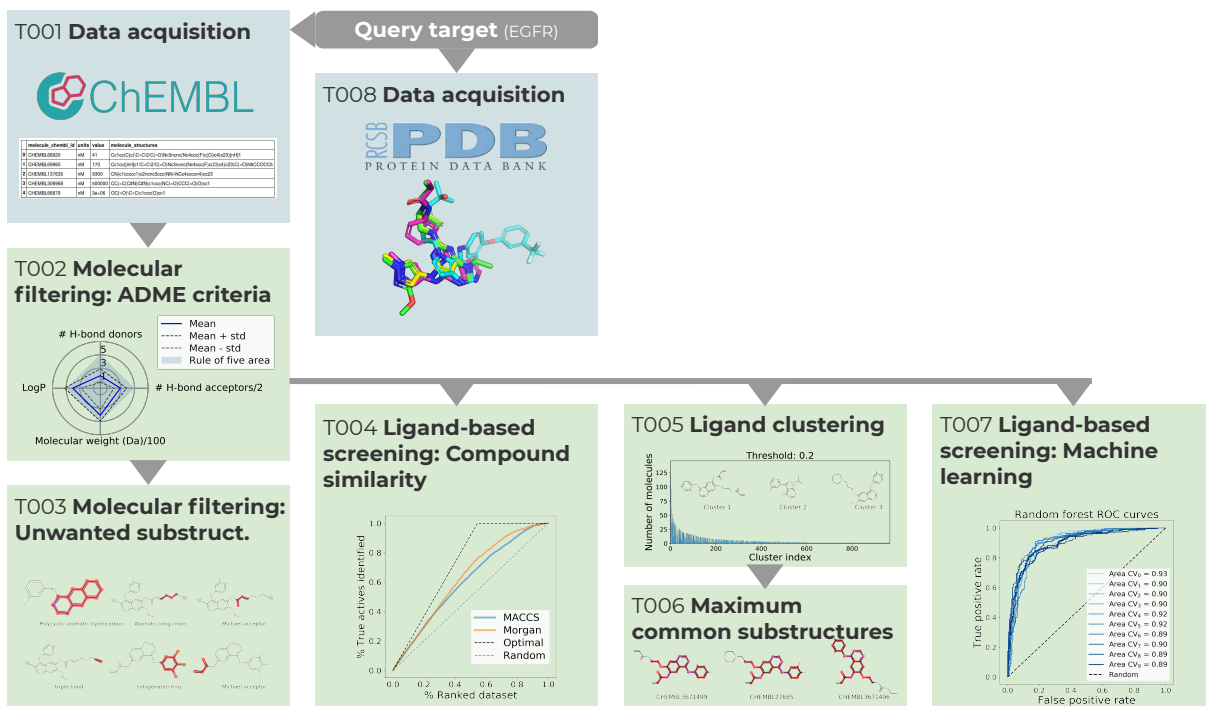
Cheminformatics | Structural Bioinformatics | Online API Webservices



Especially the cheminformatics-focused topics
are powered by the RDKit

TeachOpenCADD 2019

Cheminformatics | Structural Bioinformatics | Online API Webservices



These topics are not only available as Jupyter Notebooks but also as KNIME Workflows

Cheminformatics | Structural Bioinformatics | Online API Webservices



All tasks rely on external open source Python packages and databases/webserver

External resources

Python packages

- Cheminformatics and structural bioinformatics: `rdkit`, `openbabel`, `mdanalysis`, `biopython`, `biopandas`, `opencadd`, `plip`, `openff`, `openff-toolkit`, `openmmforcefields`, `pdbfixer`, `mdanalysis`, `biotite`, `smina`
- Data science (PyData stack): `numpy`, `pandas`, `scikit-learn`, `keras`, `jupyter`, `ipywidgets`
- Data visualization: `matplotlib`, `mpl_toolkits`, `matplotlib_venn`, `seaborn`, `nglview`
- Web services clients: `pypdb`, `chembl_webresource_client`, `requests`, `bravado`, `beautifulsoup4`
- Utilities: `tqdm`, `requests_cache`, `redo`, `google-colab`, `condacolab`
- Continuous integration: `pytest`, `nbval`
- Documentation: `sphinx`, `nbsphinx`
- Code style: `black-nb`

Databases and webservers

- ChEMBL
- RCSB PDB
- KLIFS
- PubMed
- ProteinsPlus

If we are using your resource and forgot to add it here, please contact us so that we can rectify this, thank you!

Maintaining pipelines performing diverse tasks & using many different resources can be quite a challenge!

Especially with growing content we need automatized testing!

- Nightly tests
- Notify us if notebooks throw errors
- Notify us if notebooks' output is changing / differing between operating systems



[Cookiecutter-CMS Python Packages](https://github.com/volkamerlab/teachopencadd/tree/update-readme-docs#external-resources)

<https://github.com/volkamerlab/teachopencadd/tree/update-readme-docs#external-resources>

TeachOpenCADD 2021: New website!

TeachOpenCADD (WIP 🚧)

Our talktutorials

Talktutorials by collection

Complete list of talktutorials

Run locally

Installing

Contributors

For contributors

API Documentation

External resources

Python programming introduction

Cheminformatics resources
collections

Packages and webservers used
in TeachOpenCADD

TeachOpenCADD

Open source programming packages for cheminformatics and structural bioinformatics are powerful tools to build modular, reproducible, and reusable pipelines for computer-aided drug design (CADD). While documentation for such tools is available, only few freely accessible examples are available, especially users new to the field.

TeachOpenCADD is a teaching platform developed by students for students, which provides teaching material for central CADD topics. Since we cover both computational and experimental topics, we also address students and researchers with a biological background.

For each topic, an interactive notebook is provided, which can be run locally or online. Python packages `rdkit`, `pypdb`, `biopandas`, `nglview`, and `mdanalysis`. Topics are continuously expanded and open for contributions from the community. Beyond their teaching purpose, the notebooks can be used for research and other extensions.

Explore notebooks online

Find usage instructions (Binder & conda installation)

Find information on how to contribute to TeachOpenCADD

Contents

TeachOpenCADD

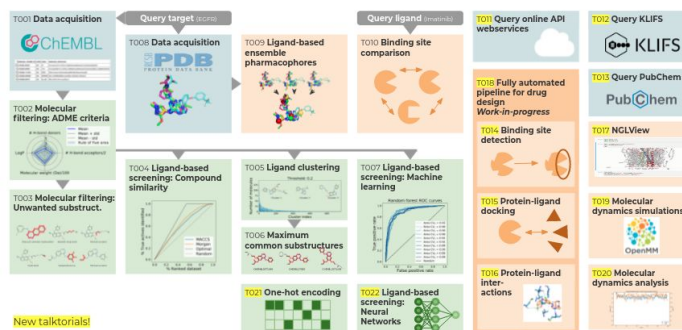
Table of contents

Citation

Funding

License

Show Source



New Talktutorials

<https://projects.volkamerlab.org/teachopencadd>

TeachOpenCADD talktorials (talk + tutorial)



T014 · Binding site detection

Search



TeachOpenCADD
254 Stars · 89 Forks

TeachOpenCADD (WIP 🚧)

Our talktorials

Talktorials by collection

Complete list of talktorials

T001 · Compound data acquisition (ChEMBL)

T002 · Molecular filtering: ADME and lead-likeness criteria

T003 · Molecular filtering: unwanted substructures

T004 · Ligand-based screening: compound similarity

T005 · Compound clustering

T006 · Maximum common substructure

T007 · Ligand-based screening: machine learning

T008 · Protein data acquisition: Protein Data Bank (PDB)

T009 · Ligand-based pharmacophores

T010 · Binding site similarity and off-target prediction

T011 · Querying online API webservices

T012 · Data acquisition from KLIFS

T013 · Data acquisition from PubChem

T014 · Binding site detection

T015 · Protein ligand docking

T016 · Protein-ligand interactions

T017 · Advanced NCI view

Binding site detection using DoGSiteScorer

We first define a function to query the server for a protein of interest. Infos on the REST API can be found [here](#).

```
[3]: def submit_dogsitescorer_job_with_pdbid(pdb_code, chain_id, ligand=""):
    """
    Submit PDB ID to DoGSiteScorer webserver using their API and get back URL for job location.

    Parameters
    -----
    pdb_code : str
        4-letter value of PDB ID, e.g. "1ABC"
    chain_id : str
        Chain ID, e.g. "A"
    ligand : str
        Name of ligand bound to PDB structure with pdb_id, e.g. "W32_A_1101".
        Currently, the ligand name must be checked manually on the DoGSiteScorer website.

    Returns
    -----
    str
        Job location URL

    References
    -----
    Function is adapted from DoGSiteScorer (https://github.com/mergpy)

    # Submit job to DoGSiteScorer
    # For details on the REST API see https://github.com/mergpy
    r = requests.post(
        "https://protein.ligand.docking.org/api/v1/submit_job",
        json={
            "dogsitescorer": {
                "pdbCode": pdb_code, # PDB code of protein
                "analysisDetail": "1", # 1 = include subpockets in results
                "bindingSitePredictionGranularity": "1", # 1 = include drugability scores
                "ligand": ligand, # if name is specified, ligand coverage is calculated
                "chain": chain_id, # if chain is specified, calculation is only performed on this chain
            }
        },
        headers={"Content-type": "application/json", "Accept": "application/json"},
    )
```

Text, code, and output all-in-one!

Motivation, theoretical background, hands-on programming, discussion & quiz

Towards coding best practices: Pythonic code style, docstrings & code comments

Contents

T014 · Binding site detection

Aim of this talktorial

Contents in Theory

Contents in Practical

References

Theory

Protein binding sites

Binding site detection

Methods overview

DoGSiteScorer

Comparison to KLIFS pocket

Practical

Binding site detection using DoGSiteScorer

Job submission for structure of interest

Get DoGSiteScorer pocket metadata

Pick the most suitable pocket

Get binding site file content

Investigate detected pocket

Comparison between DoGSiteScorer and KLIFS pocket

Get DoGSiteScorer pocket residues

Get KLIFS pocket residues

Overlap of pocket residues

Discussion

Quiz

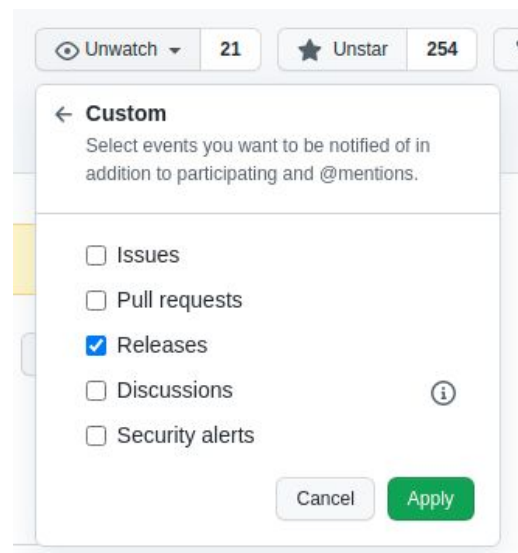
New official release coming VERY soon!

Open TODOs

- Get TeachOpenCADD on conda-forge
- Finish last pending talktorial T018 (see progress in [PR #125](#))
- Fix a few website rendering artifacts

Stay tuned

- Watch new releases



- Or follow me on Twitter for TeachOpenCADD updates

@dominiquesydow

Acknowledgements

Talktorials T001-T010

Students from CADD course 2017/18!!

Dominique Sydow
Andrea Morger
Maximilian Driller
Andrea Volkamer

KNIME workflows W1-8

Michele Wichmann
Dominique Sydow
Jaime Rodríguez-Guerra
Greg Landrum
Daria Goldmann
Andrea Volkamer

RDKit UGM 2019 Hackathon

Richard Gowers
Floriane Montanari
Jaime Rodríguez-Guerra
Dominique Sydow
Talia Kimber
Andrea Morger

TeachOpenCADD v2

Andrea Volkamer
Jaime Rodríguez-Guerra
Dominique Sydow
Talia Kimber
David Schaller
Corey Taylor
Mareike Leja
Sakshi Misra
Yonghui Chen
Michele Wichmann
Armin Ariamajd

Additional contributors

Greg Landrum
Jeff Wagner
Piedro Gerletti
Ahmed Atta
Melanie Vogel
Abishek Laxmanan Ravi Shankar
Maria Trofimova
Ross Johnstone
Thomas Holder
Andrew Wilkinson

AG Volkamer is supported by

STIFTUNG CHARITÉ



DFG Deutsche Forschungsgemeinschaft



BASF
We create chemistry



Bundesministerium
für Bildung
und Forschung



TeachOpenCADD is being used and adapted!

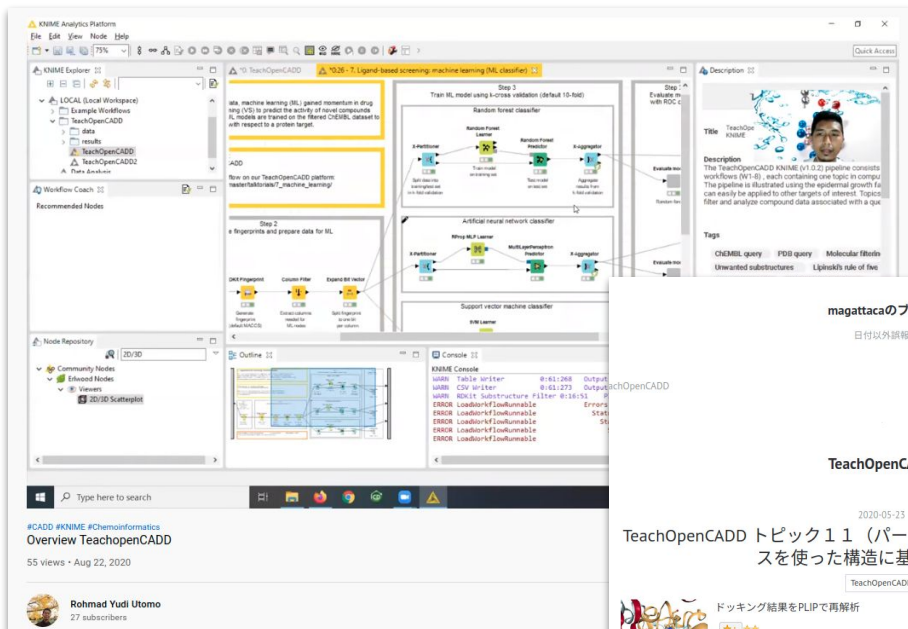
<https://github.com/volkamerlab/teachopencadd>

☆ Star

254

🍴 Fork

89



https://www.youtube.com/watch?v=AaqOaNiD_94



<https://magattaca.hatenablog.com/archive/category/TeachOpenCADD>



<https://note.com/knimesupportteam/n/n330111da83b2>