Computational Chemistry Group

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Practical Intro to the RDKit

Greg Landrum, Ph.D.

AIChemist/AIDD, 15 March 2024

greglandrum

@greg_landrum.bsky.social

@dr_greg_landrum@sciencemastodon.com

Sneak preview

After these slides, I will be working through an Jupyter notebook with a bit of tutorial and some practical details.

The notebook is here:
https://github.com/greglandrum/AICHEM_2024

Me

- RDKit principal developer
- Now:
  - Senior scientist, Riniker Lab, ETH Zurich
  - T5 Informatics GmbH
  - Senior advisor, KNIME AG
- Before:
  - KNIME (Zurich/Konstanz)
  - Novartis (Basel)
  - startups (San Francisco Bay area)

The RDKit: an open-source toolkit for cheminformatics

What *is* a toolkit anyway?

A collection of tools for building things

some simple

some not

you'll use some all the time

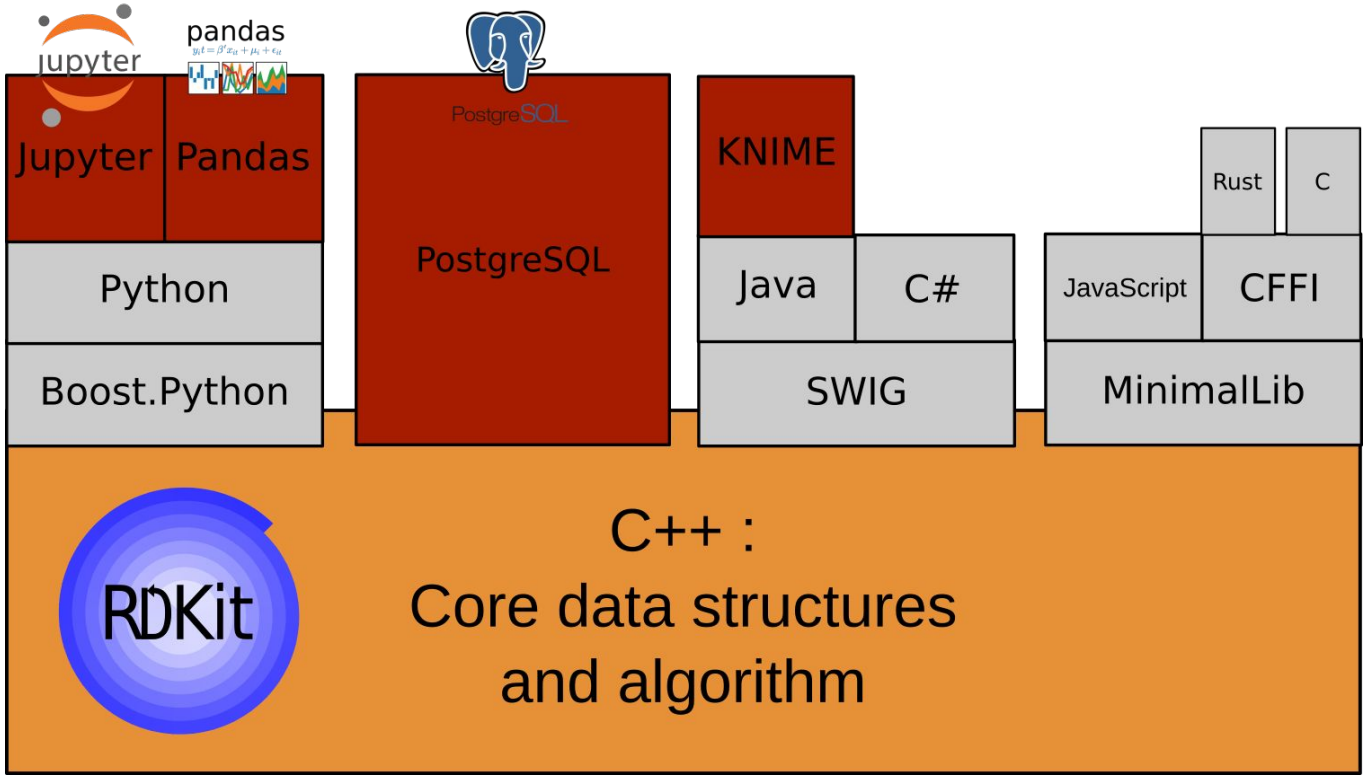
some you'll never use and may not even know about

maybe you don't even know that you're using it

An open source toolkit for cheminformatics

- Business-friendly BSD license
- Core data structures and algorithms in C++
- Python 3.x wrapper generated using Boost.Python
- Java and C# wrappers generated with SWIG
- JavaScript wrappers
- CFFI interface for usage from other languages
- 2D and 3D molecular operations
- Descriptor generation for machine learning
- Molecular database cartridge for PostgreSQL
- Cheminformatics nodes for KNIME (distributed from the KNIME community site: http://www.knime.org/rdkit)

Ecosystem



Exact same implementation regardless of where you are using it from

Details

- https://www.rdkit.org
- Supports Mac/Windows/Linux
- Major releases every 6 months, minor releases ~monthly
  - Python releases via conda-forge and pypi
  - JS releases via NPM
- Github (https://github.com/rdkit): Downloads, bug tracker, git repository, discussions
- Mailing lists at https://sourceforge.net/p/rdkit/mailman/, searchable archives available for rdkit-discuss and rdkit-devel
- Blog (https://greglandrum.github.io/rdkit-blog/): Tips, tricks, random stuff
- KNIME integration (https://github.com/rdkit/knime-rdkit): RDKit nodes for KNIME (also just from the community download site inside of KNIME)
- LinkedIn: https://www.linkedin.com/groups/8192558

Functionality[1]

- Fingerprints
- Descriptors
- Reactions
- MCS
- Enhanced stereochemistry
- Molecular standardization
- Depiction
- Diversity picking
- Tight integration with Jupyter and pandas
- Conformation generation
- 3D descriptors
- UFF and MMFF94/MMFF94S
- Open3D Align
- Feature map vectors
- Pharmacophore embedding

[1]A not-quite-random selection

Usage in other open-source projects (updated 2021)

- Shape-IT - shape-based alignment
- DockOnSurf - high-throughput code to find stable geometries for molecules on surfaces
- https://datamol.io/ - A Python library to intuitively manipulate molecules.
- Scopy - Python library for desirable HTS/VS database design
- ChEMBL Structure Pipeline - ChEMBL protocols used to standardise and salt strip molecules.
- FPSim2 - Simple package for fast molecular similarity searches.
- stk (docs, paper) - a Python library for building, manipulating, analyzing and automatic design of molecules.
- OpenFF - Open source approach for better force fields
- gpusimilarity - GPU implementation of fingerprint similarity searching
- Samson Connect - Software for adaptive modeling and simulation of nanosystems
- mol_frame - Chemical Structure Handling for Dask and Pandas DataFrames
- mmpdb 2.0 - matched molecular pair database generation and analysis
- CheTo - Chemical topic modeling
- OCEAN - web-tool for target-prediction of chemical structures which uses ChEMBL as datasource
- Coot - software for macromolecular model building, model completion and validation
- DeepChem - deep learning toolkit for drug discovery
- sdf2ppt - Reads an SDFile and displays molecules as image grid in powerpoint/openoffice presentation.
- chemfp
- PYPL - Simple cartridge that lets you call Python scripts from Oracle PL/SQL.
- WONKA - Tool for analysis and interrogation of protein-ligand crystal structures
- OOMMPPAA - Tool for directed synthesis and data analysis based on protein-ligand crystal structures
- chemicalite - SQLite integration for the RDKit
- django-rdkit - Django integration for the RDKit
- … more …

Usage in online tools/resources

- ChEMBL
- ZINC
- Google Patents
- PDBe
- Enamine
- TeachOpenCADD

Disclaimer: this info is from public statements made by people associated with those projects. I almost certainly have forgotten someone

Usage in commercial tools

- Amazon Web Services
- Collaborative Drug Discovery
- Cresset Software
- Dalke Scientific Software
- Datagrok
- Glysade
- MedChemica
- NextMove Software
- Schrödinger
- SCM
- Wolfram Research

Disclaimer: this info is from public statements made by people from those companies.
I almost certainly have forgotten someone

Documentation

Getting Started with the RDKit in Python

Important note

Beginning with the 2019.03 release, the RDKit is no longer supporting Python 2. If you need to continue using Python 2, please stick with a release from the 2018.09 release cycle.

What is this?

This document is intended to provide an overview of how one can use the RDKit functionality from Python. It's not comprehensive and it's not a manual.

If you find mistakes, or have suggestions for improvements, please either fix them yourselves in the source document (the .rst file) or send them to the mailing list: rdkit-devel@lists.sourceforge.net In particular, if you find yourself spending time working out how to do something that doesn't appear to be documented please contribute by writing it up for this document. Contributing to the documentation is a great service both to the RDKit community and to your future self.

Reading, Drawing, and Writing Molecules

RDKit

Open-Source Cheminformatics and Machine Learning

Table of Contents

Documentation

Getting Started in Python

Reading sets of molecules

Groups of molecules are read using a Supplier (for example, an rdkit.Chem.rdmolfiles.SDMolSupplier or a rdkit.Chem.rdmolfiles.SmilesMolSupplier):

```
>>> suppl = Chem.SDMolSupplier('data/5ht3ligs.sdf')
>>> for mol in suppl:
...   print(mol.GetNumAtoms())
...
20
24
24
26
```

You can easily produce lists of molecules from a Supplier:

```
>>> mols = [x for x in suppl]
>>> len(mols)
4
```

or just treat the Supplier itself as a random-access object:

```
>>> suppl[0].GetNumAtoms()
20
```

Two good practices when working with Suppliers are to use a context manager and to test each molecule to see if it was correctly read before working with it:

```
>>> with Chem.SDMolSupplier('data/5ht3ligs.sdf') as suppl:
...   for mol in suppl:
...     if mol is None: continue
...     print(mol.GetNumAtoms())
...
20
24
24
26
```

Documentation

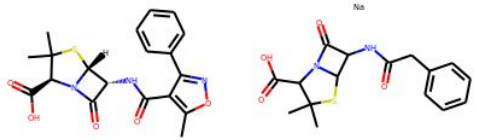
Cookbook

Black and White Molecules

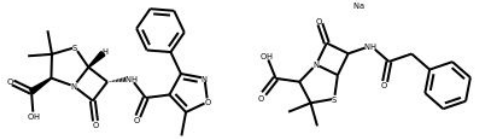
Author: Greg Landrum and Vincent Scalfani
Source: https://gist.github.com/greglandrum/d85d5693e57c306e30057ec4d4d11342 and https://github.com/rdkit/rdkit/discussions/5885
Index ID#: RDKitCB_1
Summary: Draw a molecule in black and white.

```
from rdkit import Chem
from rdkit.Chem.Draw import IPythonConsole
from rdkit.Chem import Draw
```

```
ms = [Chem.MolFromSmiles(x) for x in ('Cc1onc(-c2ccccc2)c1C(=O)N[C@@H]1C(=O)N2[C@@H](C(=O)O)C(C)
Draw.MolsToGridImage(ms)
```



```
IPythonConsole.drawOptions.useBWAtomPalette()
Draw.MolsToGridImage(ms)
```



Documentation

Reference

```
>>> Chem.MolFromSmiles('CC(=O)OC').GetSubstructMatches(Chem.MolFromSmarts('[z{1-}]'))
((1,), (4,))
>>> Chem.MolFromSmiles('CC(=O)OC').GetSubstructMatches(Chem.MolFromSmarts('[D{2-3}]'))
((1,), (3,))
>>> Chem.MolFromSmiles('CC(=O)OC.C').GetSubstructMatches(Chem.MolFromSmarts('[D{-2}]'))
((0,), (2,), (3,), (4,), (5,))
```

SMARTS Reference

Note that the text versions of the tables below include some backslash characters to escape special characters. This is a wart from the documentation system we are using. Please ignore those characters.

Atoms

Primitive	Property	"Default value"	Range?	Notes
a	"aromatic atom"			
A	"aliphatic atom"			
d	"non-hydrogen degree"	1	Y	extension
D	"explicit degree"	1	Y	
h	"number of implicit hs"	>0	Y	
H	"total number of Hs"	1		
r	"size of smallest SSSR ring"	>0	Y	
R	"number of SSSR rings"	>0	Y	
v	"total valence"	1	Y	
x	"number of ring bonds"	>0	Y	
X	"total degree"	1	Y	
z	"number of heteroatom neighbors"	>0	Y	extension
Z	"number of aliphatic heteroatom neighbors"	>0	Y	extension

Community

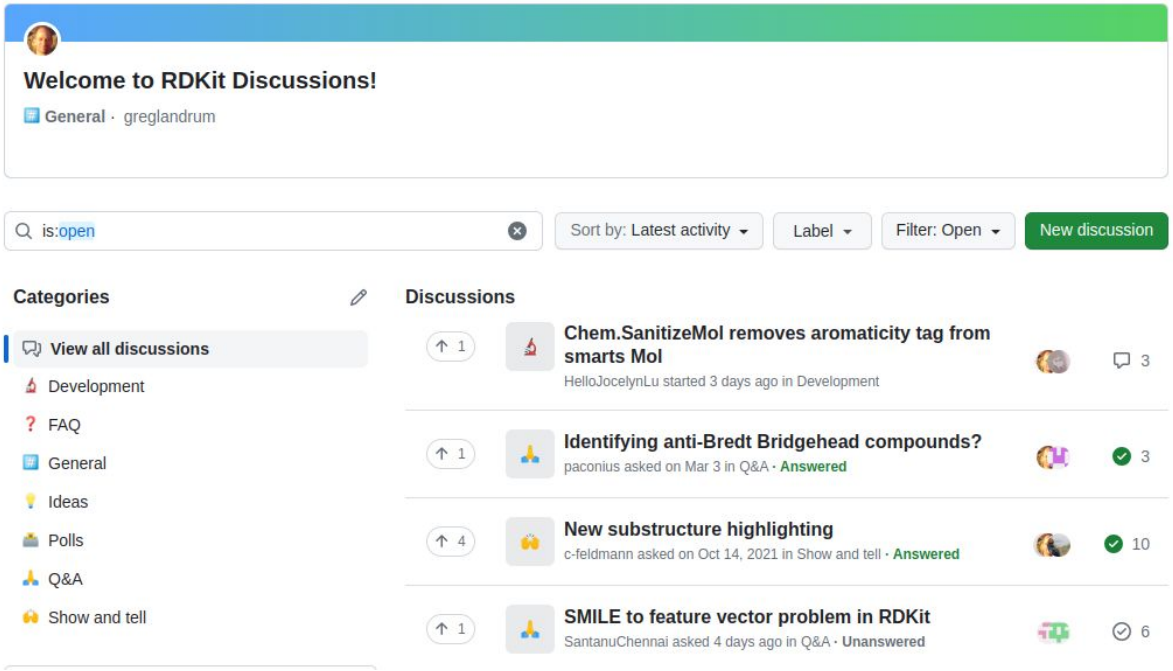
The heart of any *successful* open-source project

Support

- Web searches
- Mailing list
- Github discussions

- Commercial support

Community support



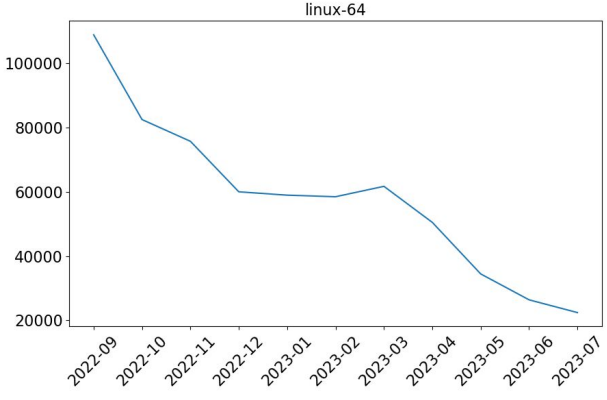
Adoption / usage

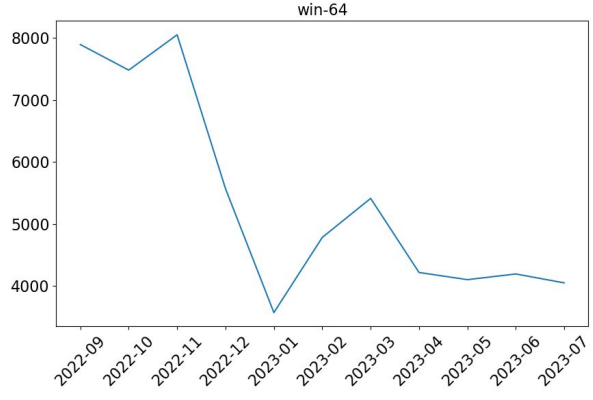
Unlike with web apps or commercial software, this is tricky to figure out with open source tools, but let's try.

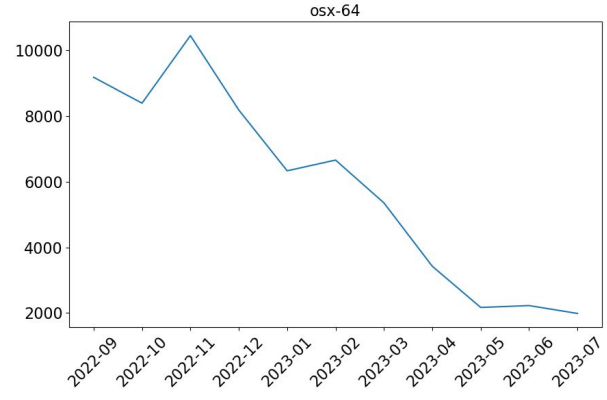
Other adoption measures

- Mailing lists: ~250 messages to rdkit-discuss from 2022.09 - 2023.08
- Google scholar: >2300 hits for "rdkit" in 2022, >2000 so far in 2023
- Searching github for `"from rdkit import Chem"` returns >27000 code results
- Each of the last ten in-person UGMs at capacity with 40-150 attendees

Usage: Conda install counts (by operating system)

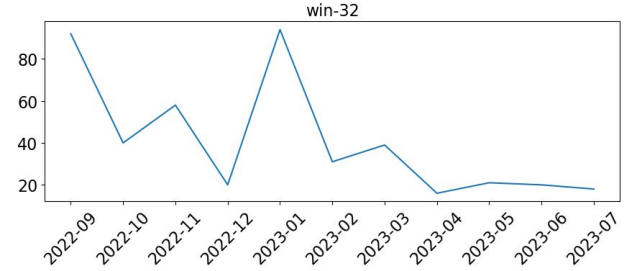


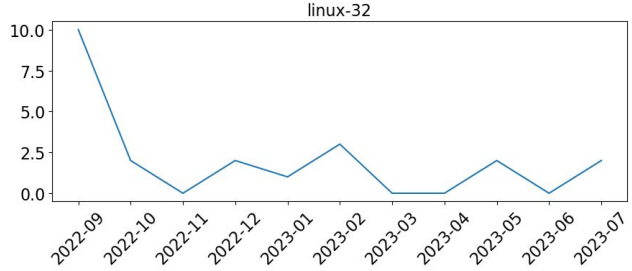


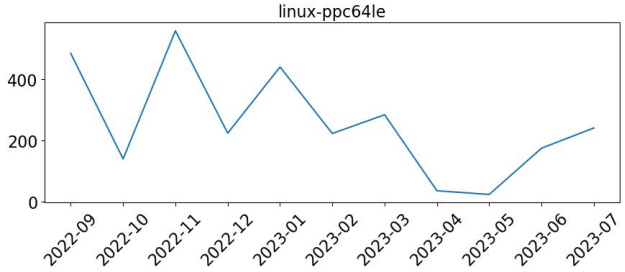


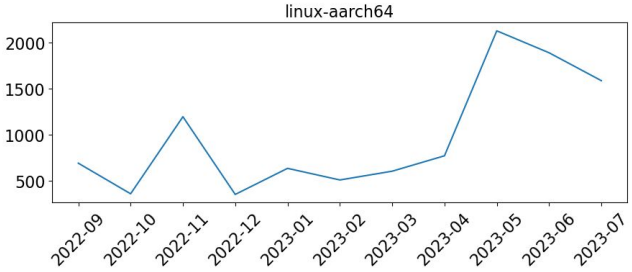
Last 12 months
Data collected using the `condastats` package

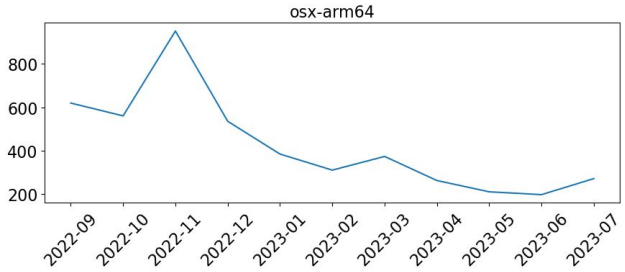
Usage: Conda install counts (by operating system)





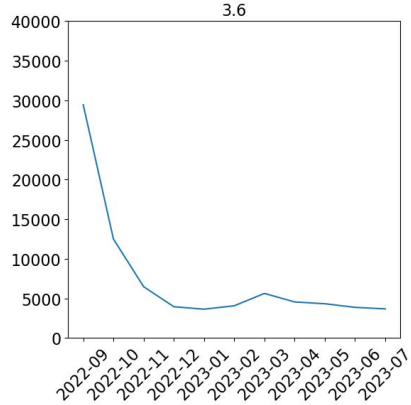


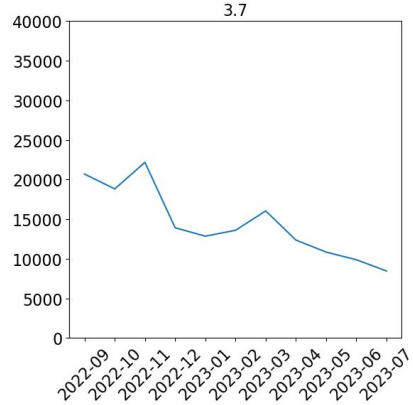


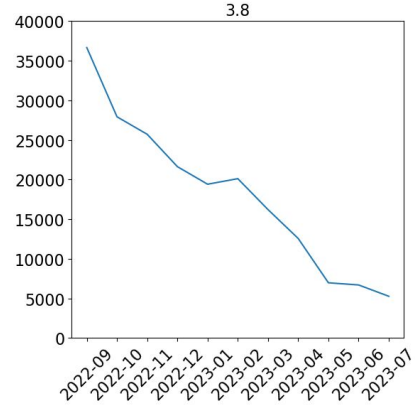


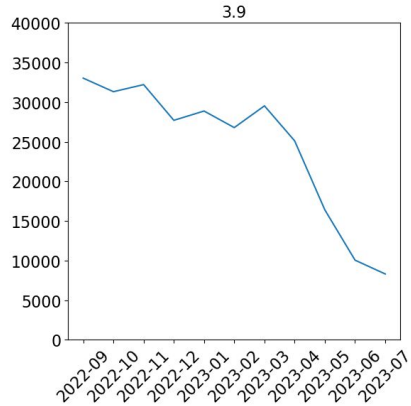
Less common operating systems / hardware combosd

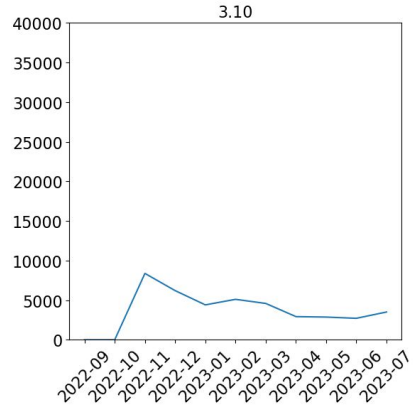
Usage: Conda install counts (by python version)





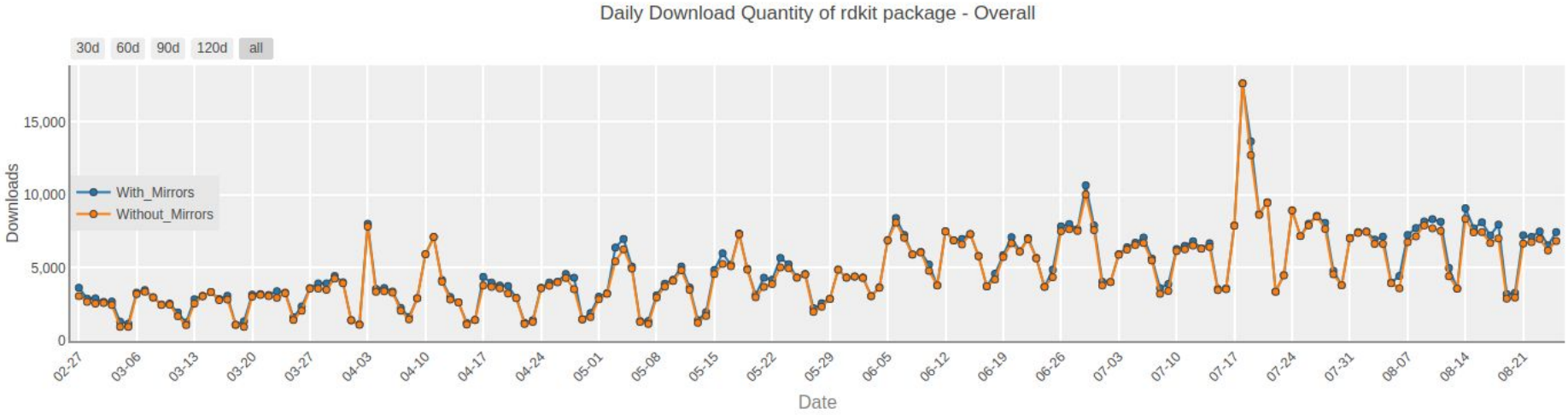






v3.11 was not available from condastats when I ran these queries

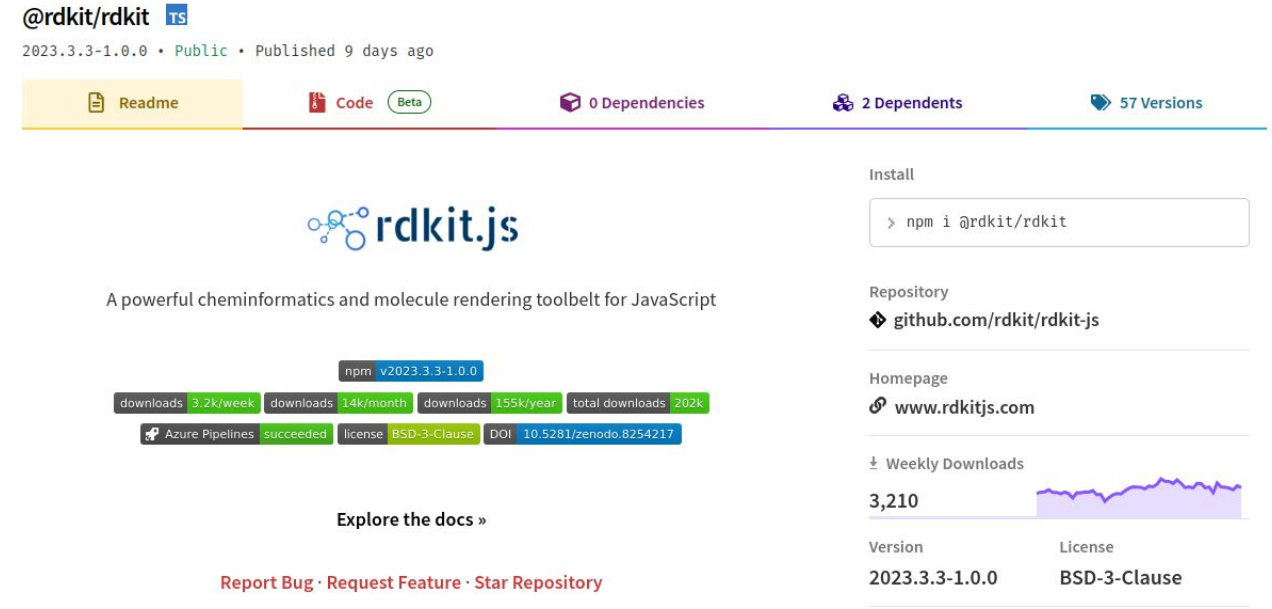
Usage: PyPi



Thanks to Chris Kuenneth for getting the pypi installs set up!

Last 120 days of data from https://pypistats.org/packages/rdkit-pypi

rdkit-js usage:



Thanks to Michel Moreau for getting this set up!

Roadmap

Future work tends to be determined by what's needed for active projects or requests that come out of the community. So there's not much of a roadmap.

Still, some parts of the way forward are pretty obvious…

Making sure all the pieces required to build a good compound registration system are there

Making sure all the pieces required to build a good corporate chemical database are there

Better support for polymers and organometallics

Performance improvements

Ongoing improvements to the conformer generator

Ongoing refactoring and code cleanup

Taking big steps forward…

Some things are hard…

Technology changes (i.e. taking advantage of new C++ or Python versions) is tricky: which operating systems/compilers are people using?

Is it safe to remove old code that seems peripheral or redundant with functionality provided better by other packages?

There are some larger API changes to clean up old mistakes and improve performance and safety that it would be nice to make.

We really, really want to avoid the Python 2/Python 3 situation, so we can't just make arbitrary changes.

… what we're doing about it

Try to minimize hard external dependencies

Be conservative about language versions/features

Announce deprecations at least one major release in advance

"Backwards incompatible changes" doc

Version-compatibility report (for commercial support customers)

Changing the RDKit release model

Motivation: make new functionality available sooner

Previous:

- Feature releases twice a year, e.g. **2023.03**
  - Possibly including backwards-incompatible changes
- Patch releases every 4-6 weeks, e.g. 2023.03**.2**
  - Only bug fixes, but these can still change results

Changing the RDKit release model

Motivation: make new functionality available sooner

Previous:

- Feature releases twice a year, e.g. **2023.03**
  - Possibly including backwards-incompatible changes
- Patch releases every 4-6 weeks, e.g. 2023.03**.2**
  - Only bug fixes, but these can still change results

Current (as of 2023.09):

- Major releases twice a year, e.g. **2023.09**
  - Possibly including backwards-incompatible changes
- Minor releases every 4-6 weeks, e.g. 2023.09**.2**
  - Include bug fixes (can change results)
  - Include backwards-compatible new features

Acknowledgements

- Everyone who has contributed code, questions, answers, bug reports, etc
- People who have funded RDKit development (directly or indirectly)
- The others in our community who've been pushing the idea and adoption of open source

Thanks!