

Interactive rendering of 2D structures with RDKit

Paolo Tosco, Nico Pulver, Nik Stiefl

11th RDKit UGM

12 October 2022

Agenda

Where we were (2021)

Where we are (2022)

What next



Where we were (2021)

Visualize 2D molecules: from server-based...

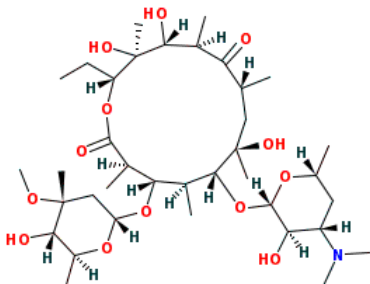
- Several apps (Python, Java, JS, .NET) @NIBR need 2D structure visualization
- This used to be handled through a web service based on the Avalon depicter
- The client requested a 2D layout to the server (SMILES or CTAB)
- The server generated the 2D layout as a PNG image and exposed it to the client through a URL
- Several configuration options were available
 - Width, height
 - Align the layout to a scaffold (SMILES or CTAB)
 - Font, coloring, etc.

...to client-based

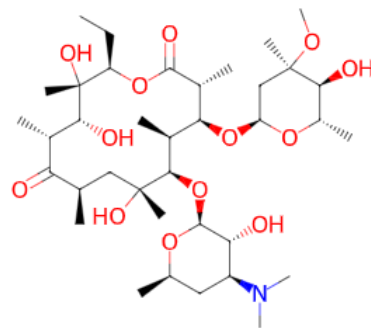
- Shut down the Avalon Depicter web service
- Generate all depictions in the clients (Python, JS, Java, .NET) using RDKit

```

```

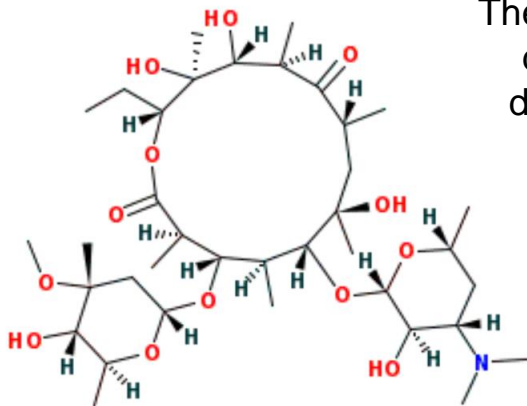


```
<div class="rdk-str-rnr-mol-container" id="rdk-str-rnr-mol-1"
data-width="300" data-height="250" data-
mol="CC[C@@H]1[C@@H]([C@H](C(=O)[C@@H](C[C@@]([C@H]([C@H]([C@H]([C@H](C(=O)O1)C)O[C@H]2C[C@@]([
C@H]([C@H](O2)C)O)(C)OC)C)O[C@H]3[C@@H]([C@H](C[C@H](O3)C)N
(C)C)O)(C)O)C)O)(C)O&w=300&h=250"></div>
```

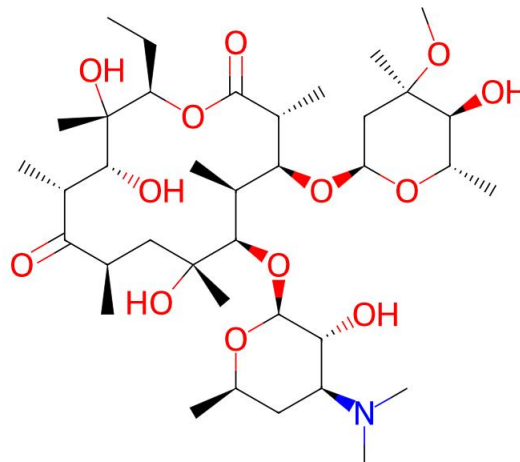


From PNG to scalable vector images

- Move from a static PNG image to a vector image (HTML5 canvas, SVG)



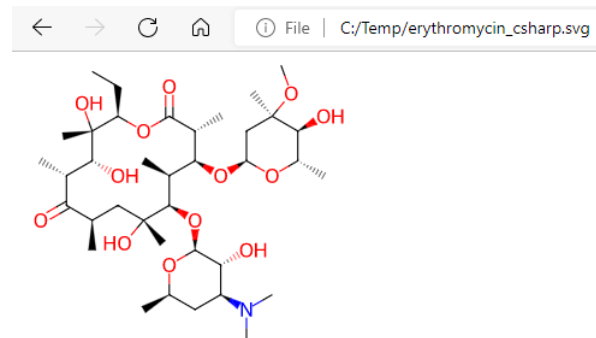
The SVG structure
does not lose
definition when
zooming in



RDKit, RDKit everywhere

- All the base functionality is available in the RDKit C++ core
- Can be accessed from any language RDKit has wrappers for (C, Python, Java, JS, .NET)

```
ROMol erythromycin =  
RWMol.MolFromSmiles("CC[C@@H]1[C@@]([C@@H]([C@H](C(=O)[C@@H](C[C@@]([C@  
@H]([C@H]([C@@H]([C@H](C(=O)O1)C)O[C@H]2C[C@@]([C@H]([C@@H](O2)C)O)(C)O  
C)C)O[C@H]3[C@@H]([C@H](C[C@H](O3)C)N(C)C)O)(C)O)C)O)(C)O");  
using (MolDraw2DSVG molDraw = new MolDraw2DSVG(300, 250)) {  
    RDKFuncs.setPreferCoordGen(true);  
    erythromycin.compute2DCoords();  
    RDKFuncs.setPreferCoordGen(false);  
    erythromycin.normalizeDepiction();  
    erythromycin.straightenDepiction();  
    RDKFuncs.updateDrawerParamsFromJSON(molDraw, drawOptions);  
    molDraw.drawMolecule(erythromycin);  
    molDraw.finishDrawing();  
    string svg = molDraw.getDrawingText();  
    File.WriteAllText("C:/Temp/erythromycin_csharp.svg", svg);  
}
```



Interactive rendering of 2D structures in the browser

- At the 2021 RDKit UGM I presented a PoC JS library to do interactive rendering of 2D structures in the browser

The screenshot displays a web browser window at <https://localhost:9900/lab>. The interface includes a top menu bar (File, Edit, View, Run, Kernel, Git, Tabs, Settings, Help) and a left sidebar with a file explorer showing a directory with 'snippets', 'StructRenderer.ipynb', and 'Untitled.ipynb'. The main workspace contains a Jupyter notebook titled 'StructRenderer.ipynb' with a code cell. To the right of the code cell is a 'Structure Renderer Options' dialog box with the following settings:

- ☐ Re-compute 2D layout
- ☒ Align to scaffold
- ☒ Highlight scaffold
- ☐ Abbreviate groups
- ☐ Add atom indices
- > Formats

Below the code cell, three chemical structures are rendered:

- A long-chain alkyl amine derivative (top right).
- A complex polycyclic molecule with a carboxylic acid group and a sulfonate group (bottom center).
- A quinoline derivative (bottom right).

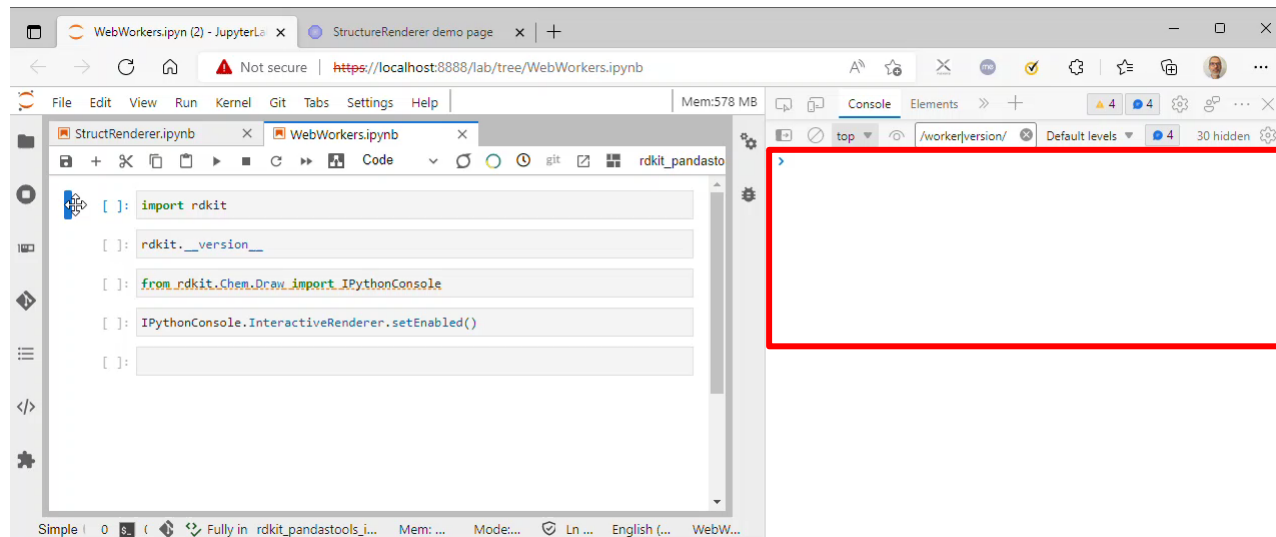
Chemical identifiers are visible next to the structures: CHEMBL1782626 and CHEMBL2106827.



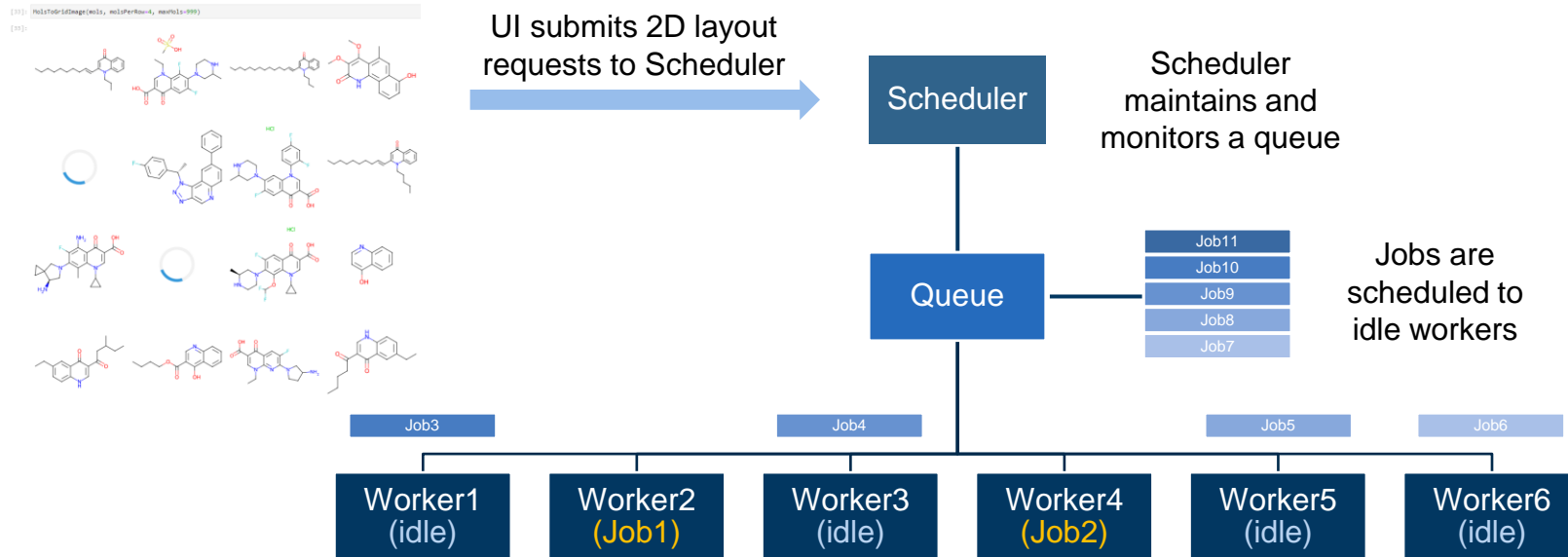
Where we are (2022)

Web Workers!

Web
Workers
are started
as soon as
rdkit-
structure-
renderer
loads

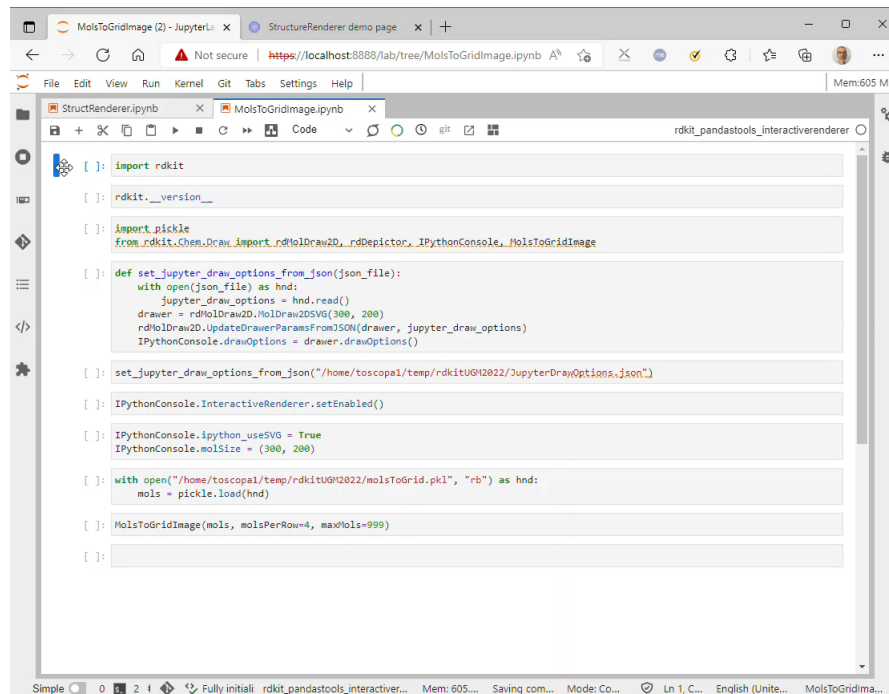


2D layout jobs are submitted to Web Workers



Long running jobs are not an issue

- The UI remains responsive also in the presence of long running jobs as there are multiple Web Workers
- Structures are displayed asynchronously



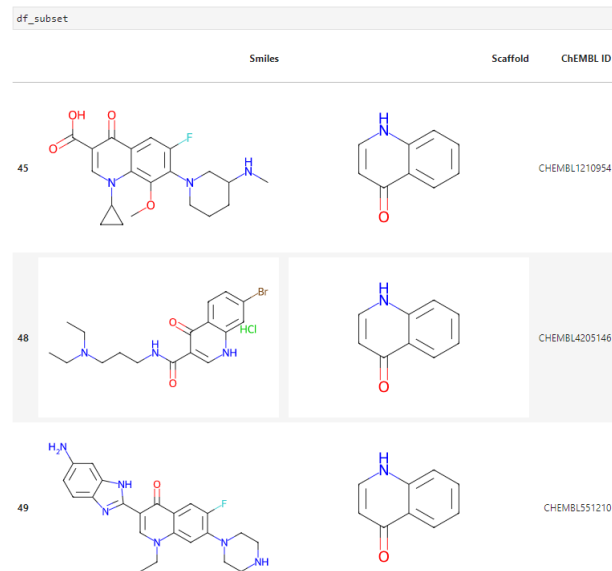
```
[ ]: import rdkit
[ ]: rdkit.__version__
[ ]: import pickle
from rdkit.Chem.Draw import rdMolDraw2D, rdDepictor, IPythonConsole, MolsToGridImage
[ ]: def set_jupyter_draw_options_from_json(json_file):
    with open(json_file) as hnd:
        jupyter_draw_options = hnd.read()
        drawer = rdMolDraw2D.MolDraw2DSVG(300, 200)
        rdMolDraw2D.UpdateDrawerParamsFromJSON(drawer, jupyter_draw_options)
        IPythonConsole.drawOptions = drawer.drawOptions()
[ ]: set_jupyter_draw_options_from_json("/home/toscopai/temp/rdkitUGH2022/JupyterDrawOptions.json")
[ ]: IPythonConsole.InteractiveRenderer.setEnabled()
[ ]: IPythonConsole.ipython_useSVG = True
IPythonConsole.molSize = (300, 200)
[ ]: with open("/home/toscopai/temp/rdkitUGH2022/molstoGrid.pkl", "rb") as hnd:
    mols = pickle.load(hnd)
[ ]: MolsToGridImage(mols, molsPerRow=4, maxMols=999)
[ ]:
```

Scaffold alignment (1)

- mol.generate_aligned_cords(scaffold, opts)

This MinimalLib function generates a 2D layout for mol by rebuilding side chains *ex novo* around the constrained scaffold

- Based on
RDDepict::generateDepictionMatching2DStructure

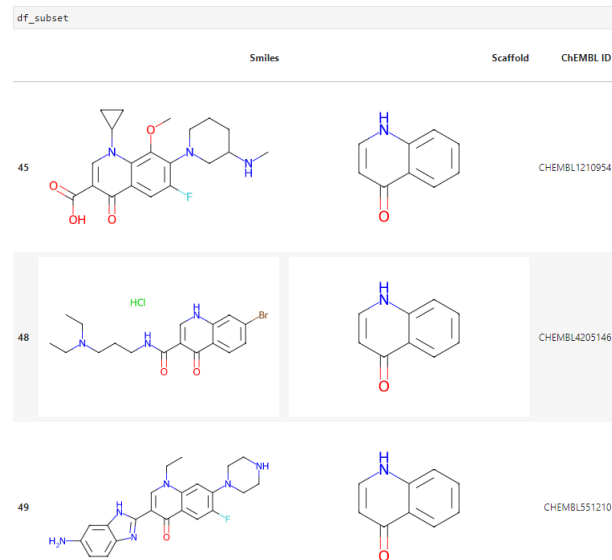


Scaffold alignment (2)

- JSmol.generate_aligned_cords(scaffold, opts)

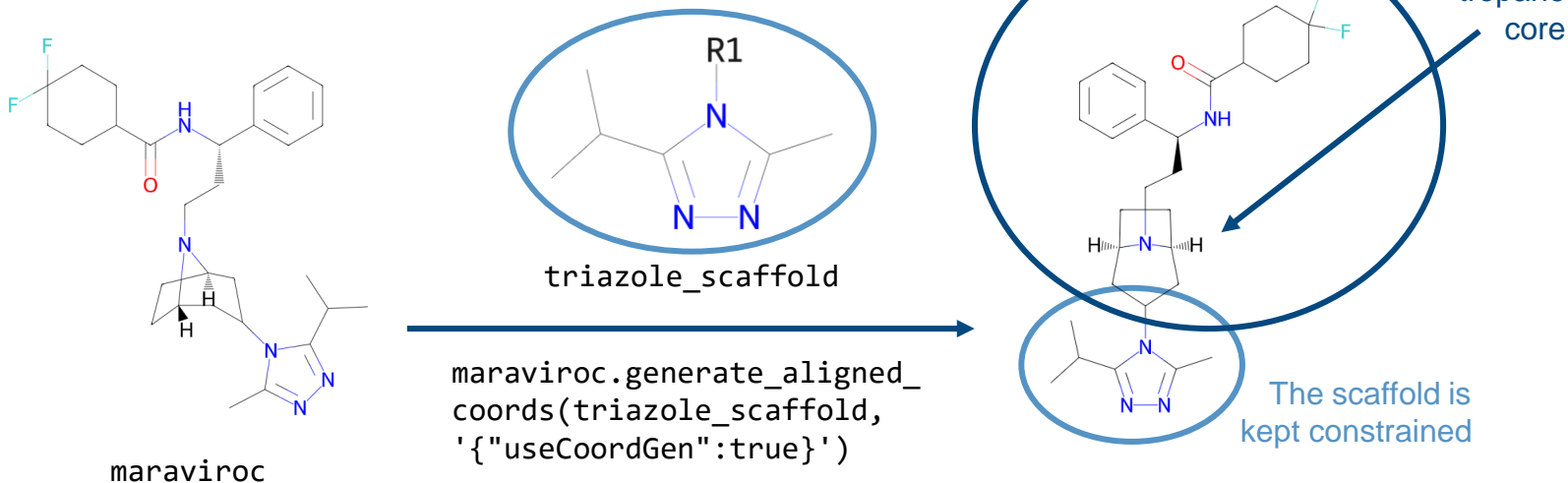
This MinimalLib function generates a 2D layout for mol by rebuilding side chains *ex novo* around the constrained scaffold

- Based on
RDDepict::generateDepictionMatching2DStructure
- Molecules in a chemical series can be oriented consistently



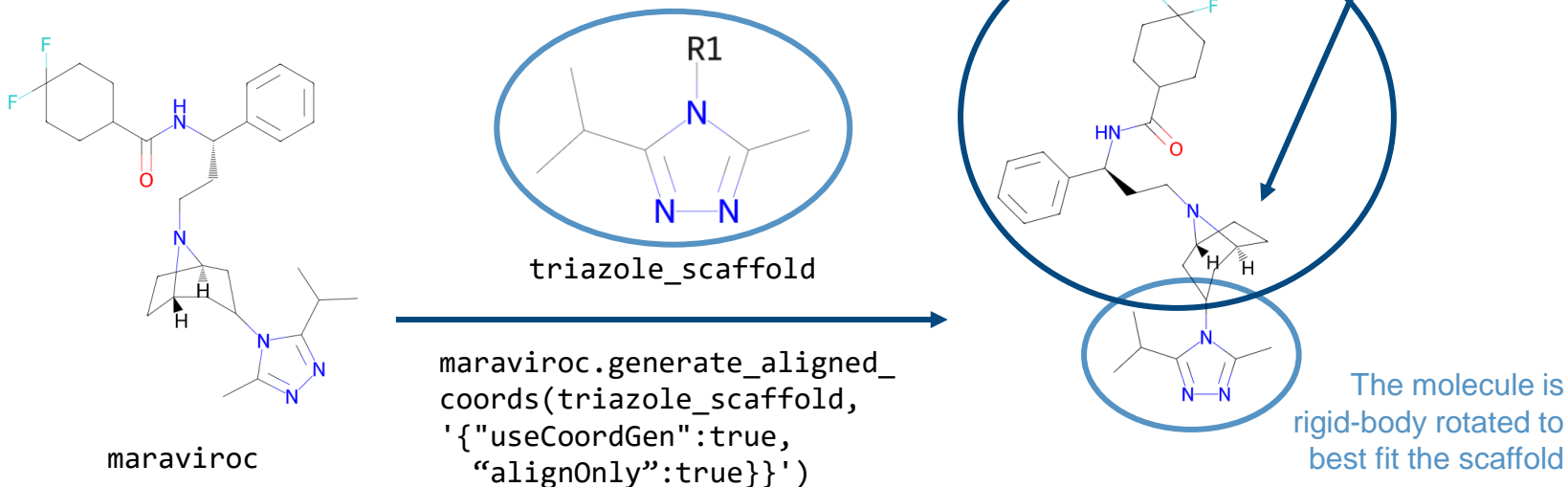
Scaffold alignment (3)

- Rebuilding coordinates *ex novo* may alter the way sidechains were originally drawn

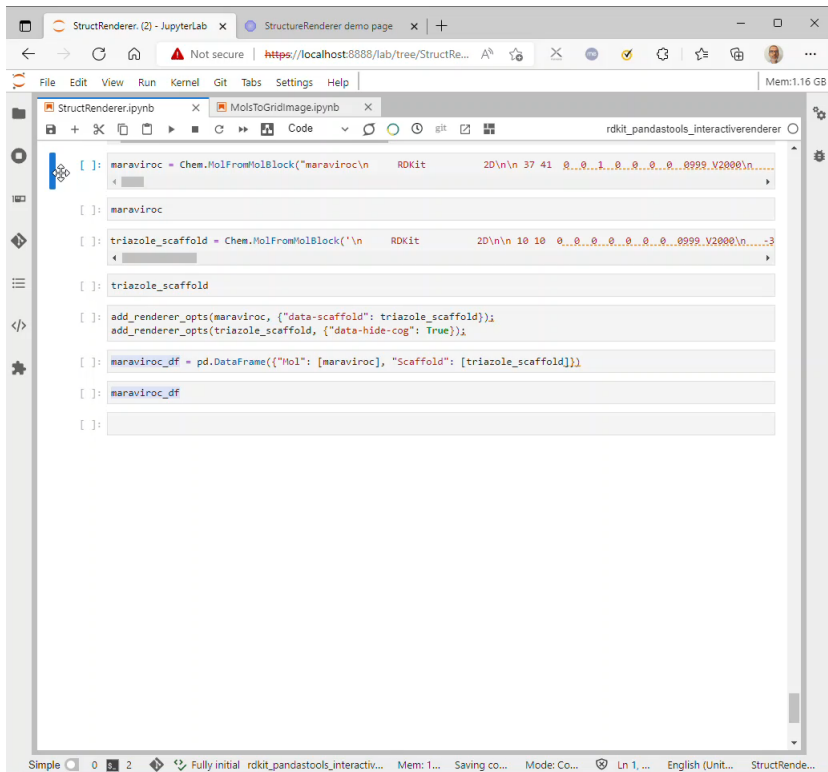


Conservative scaffold alignment (1)

- JSmol.generate_aligned_cords(scaffold, opts)
now supports an alignOnly mode



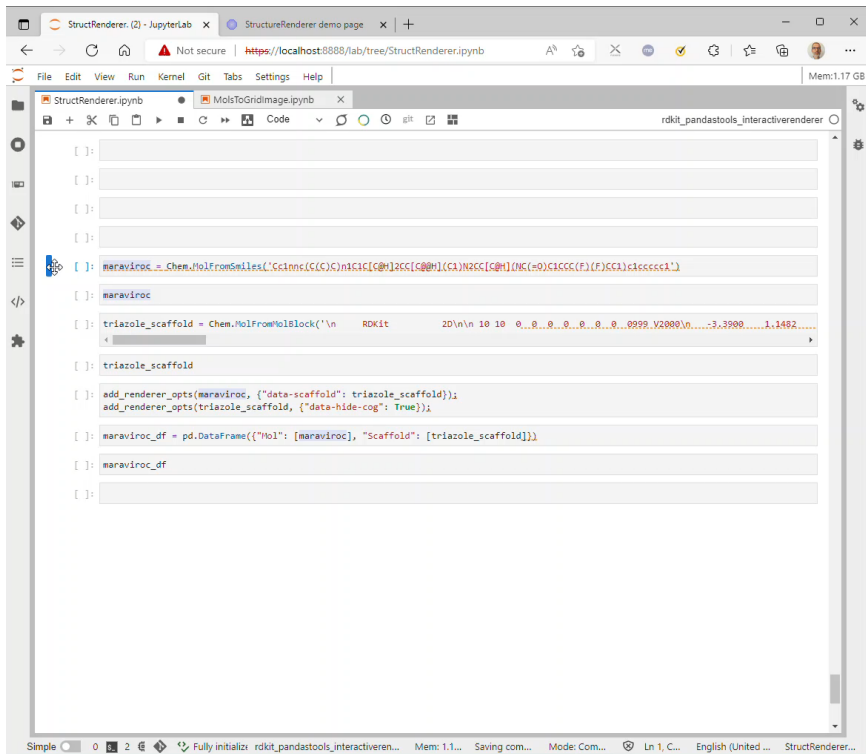
Conservative scaffold alignment (2)



```
[ ]: maraviroc = Chem.MolFromMolBlock("maraviroc\n RDKit 2D\n\n 37 41  0...0...1...0...0...0...0999.V2000\n< ...>\n[ ]: maraviroc\n[ ]: triazole_scaffold = Chem.MolFromMolBlock(''\n RDKit 2D\n\n 10 10  0...0...0...0...0...0999.V2000\n< ...>\n[ ]: triazole_scaffold\n[ ]: add_renderer_opts(maraviroc, {"data-scaffold": triazole_scaffold});\nadd_renderer_opts(triazole_scaffold, {"data-hide-cog": True});\n[ ]: maraviroc_df = pd.DataFrame({"Mol": [maraviroc], "Scaffold": [triazole_scaffold]})\n[ ]: maraviroc_df\n[ ]:
```

- rdkit-structure-renderer will do a conservative scaffold alignment unless explicitly asked to rebuild coordinates *ex novo*

Non-conservative scaffold alignment



- If the molecule does not have original coordinates, the “Recompute 2D” option is greyed out, since coordinates are anyway rebuilt *ex novo*

NIBR web apps: Intuence Discovery

Intuence Discovery **CHEMBL disubstituted indoles** Last saved 04 Oct 2022 10:26 PM CEST

Id	Structure	Primary Scaffold	R2	R5
1 CHEMBL100252		R5 	*2 	*5
2 CHEMBL100872		R5 	*2 	*5
3 CHEMBL101891		R5 	*2 	*5
4 CHEMBL106445		R5 	*2 	*5
5 CHEMBL108177		R5 	*2 	*5
6 CHEMBL1082786		R5 	*2 	*5

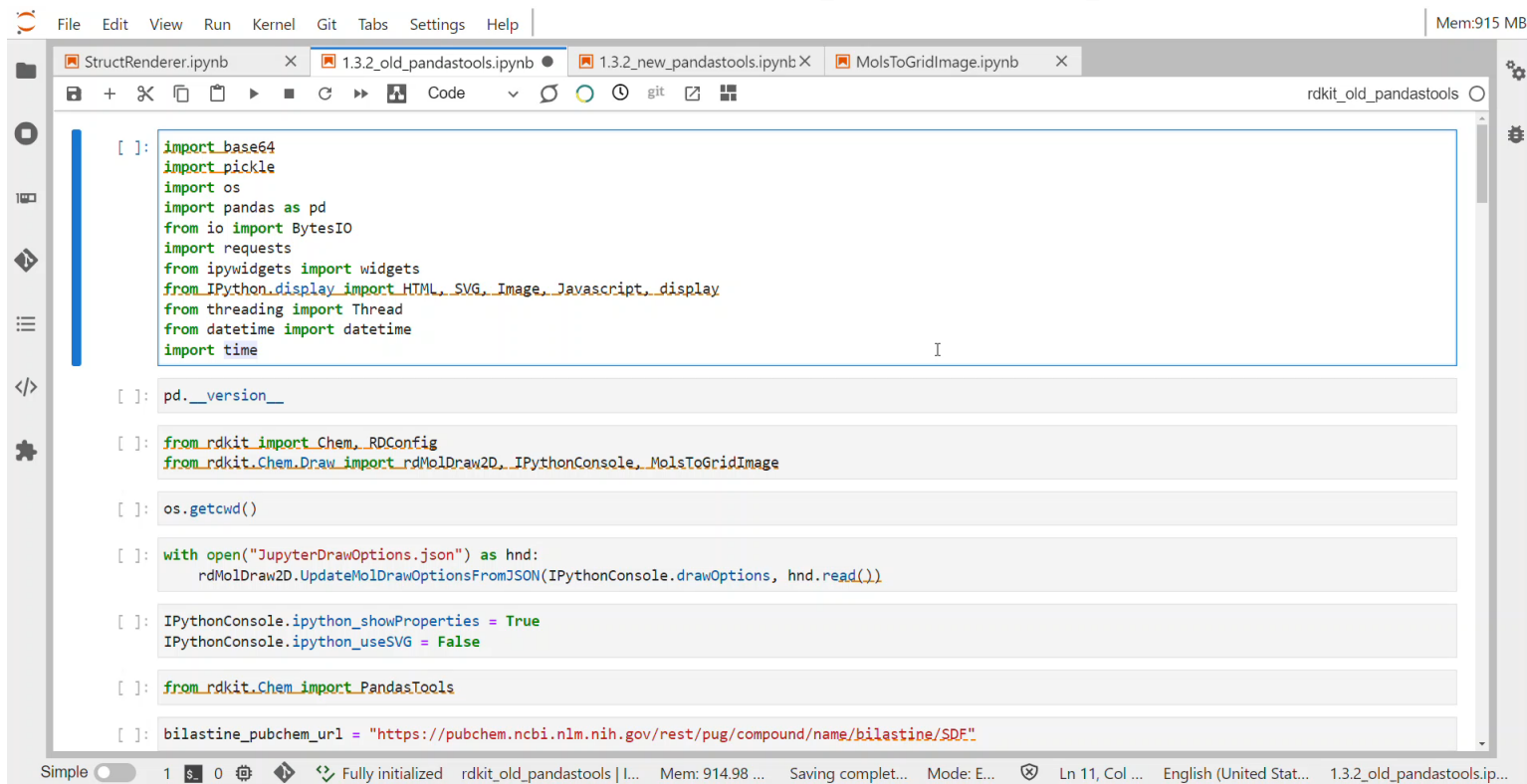
General view

ID CHEMBL106445 CHEMBL108177 CHEMBL1082786
 HT Solubility pH6.8 [mM]
 LE-MDCK v2 PappAtoB
 MDCK-MDR1 ER
 Direct LogP
 Direct LogD
 rLM CLint
 hLM CLint
 hERG Binding IC50
 NIBR logD 1.23744 3.21537 3.12380

ID CHEMBL1083614 CHEMBL1083616 CHEMBL1083942
 HT Solubility pH6.8 [mM]
 LE-MDCK v2 PappAtoB
 MDCK-MDR1 ER
 Direct LogP
 Direct LogD
 rLM CLint

+ < > ≡ MAIN 148 cols 1,736 rows 1736 selected

PandasTools issues (minor)



The screenshot shows a JupyterLab environment with a code editor and a terminal. The code editor contains a Python script for setting up PandasTools. The terminal shows the output of the script, including the version of Pandas and the location of the PandasTools module.

```
[ ]: import_base64
import_pickle
import os
import pandas as pd
from io import BytesIO
import requests
from ipywidgets import widgets
from IPython.display import HTML, SVG, Image, Javascript, display
from threading import Thread
from datetime import datetime
import time

[ ]: pd.__version__

[ ]: from_rdkit import Chem, RDConfig
from_rdkit.Chem.Draw import rdMolDraw2D, IPythonConsole, MolsToGridImage

[ ]: os.getcwd()

[ ]: with open("JupyterDrawOptions.json") as hnd:
    rdMolDraw2D.UpdateMolDrawOptionsFromJSON(IPythonConsole.drawOptions, hnd.read())

[ ]: IPythonConsole.ipython_showProperties = True
IPythonConsole.ipython_useSVG = False

[ ]: from_rdkit.Chem import PandasTools

[ ]: bilastine_pubchem_url = "https://pubchem.ncbi.nlm.nih.gov/rest/pug/compound/name/bilastine/SDF"
```

Simple 1 \$ 0 Fully initialized rdkit_old_pandastools | l... Mem: 914.98 ... Saving complet... Mode: E... Ln 11, Col ... English (United Stat... 1.3.2_old_pandastools.ip...

Jupyter Notebook performance

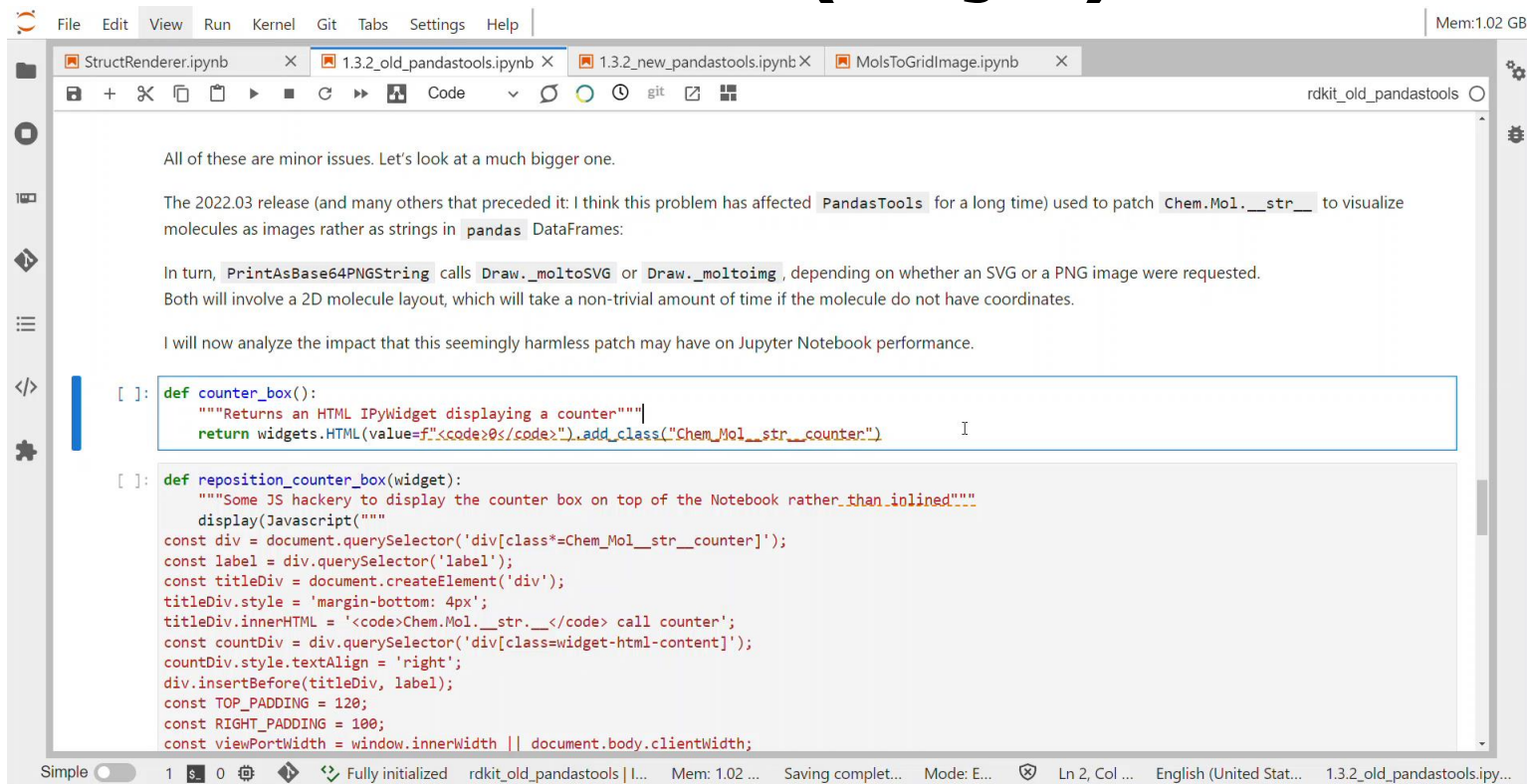
- I have had the feeling for a while that Jupyter Notebooks become sluggish after some uptime
- Google did not provide many relevant hits on that
- I was under the impression that this was happening in particular when working with pandas DataFrames
- And my DataFrames always have molecules
- Is PandasTools the culprit?

How PandasTools used to work

```
... 469     if renderer == 'String':  
470         Chem.Mol.__str__ = PrintDefaultMolRep  
471     else:  
472         Chem.Mol.__str__ = PrintAsBase64PNGString  
473     if frame is not None:  
474         frame.to_html = types.MethodType(patchPandasHTMLrepr, frame)  
475         if defPandasRepr is not None:  
476             frame._repr_html_ = types.MethodType(defPandasRepr, frame)
```

- `Chem.Mol.__str__` is redirected to `PrintAsBase64PNGString` such that molecules are visualized as PNG or SVG images rather than as strings (e.g., `<rdkit.Chem.rdchem.Mol object at 0x7fa6bc5441c0>`)
- This means that a 2D layout may take place at each `Chem.Mol.__str__` call

PandasTools issues (major)



The screenshot shows a Jupyter Notebook with a menu bar (File, Edit, View, Run, Kernel, Git, Tabs, Settings, Help) and a memory indicator (Mem: 1.02 GB). The notebook has four tabs: StructRenderer.ipynb, 1.3.2_old_pandastools.ipynb (active), 1.3.2_new_pandastools.ipynb, and MolsToGridImage.ipynb. The active tab contains the following text and code:

All of these are minor issues. Let's look at a much bigger one.

The 2022.03 release (and many others that preceded it: I think this problem has affected PandasTools for a long time) used to patch Chem.Mol.__str__ to visualize molecules as images rather as strings in pandas DataFrames:

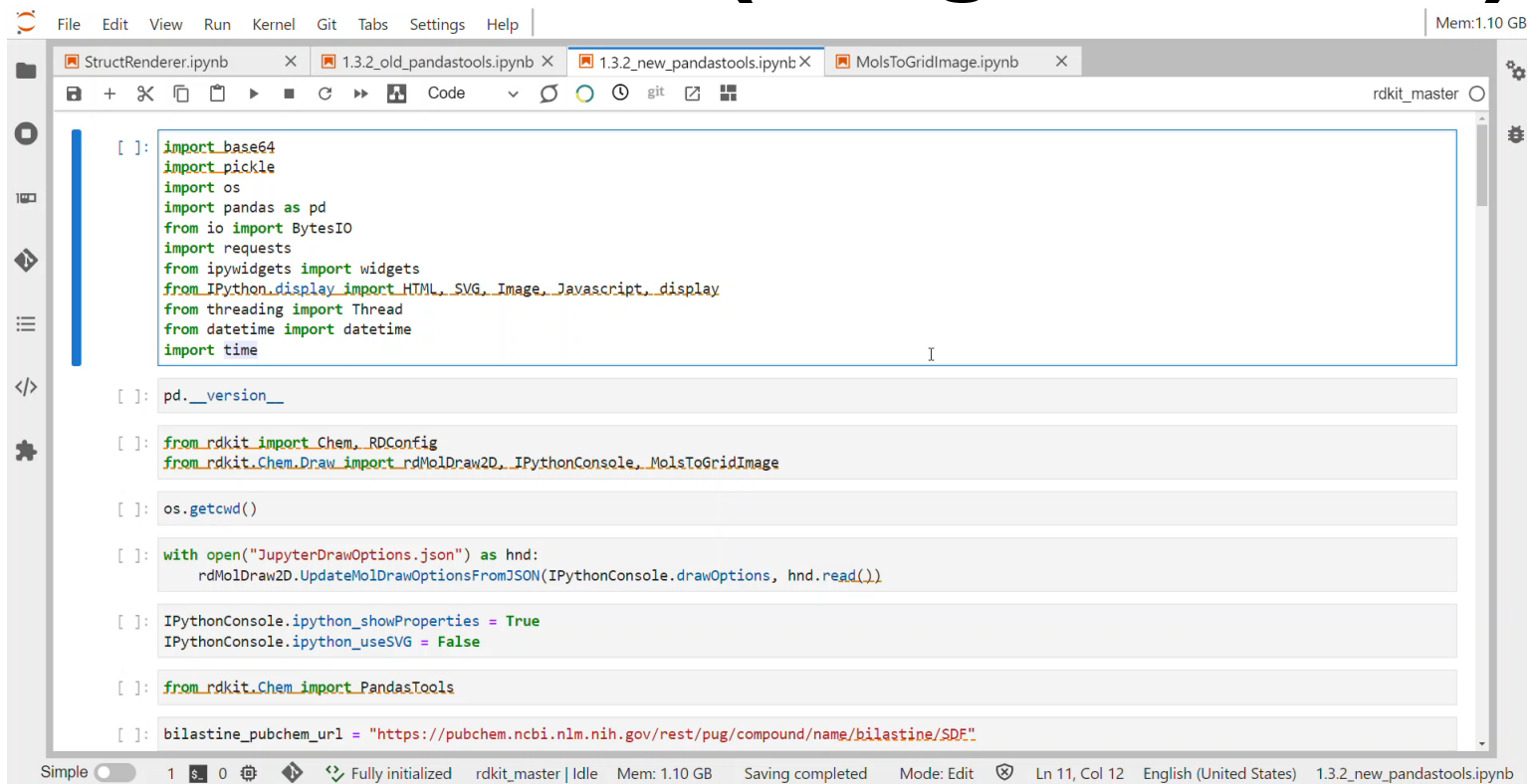
In turn, PrintAsBase64PNGString calls Draw._moltoSVG or Draw._moltoimg, depending on whether an SVG or a PNG image were requested. Both will involve a 2D molecule layout, which will take a non-trivial amount of time if the molecule do not have coordinates.

I will now analyze the impact that this seemingly harmless patch may have on Jupyter Notebook performance.

```
[ ]: def counter_box():
    """Returns an HTML IPyWidget displaying a counter"""
    return widgets.HTML(value=f"<code>{0}</code>").add_class("Chem_Mol__str__counter")

[ ]: def reposition_counter_box(widget):
    """Some JS hackery to display the counter box on top of the Notebook rather than inlined"""
    display(Javascript("""
const div = document.querySelector('div[class*=Chem_Mol__str__counter]');
const label = div.querySelector('label');
const titleDiv = document.createElement('div');
titleDiv.style = 'margin-bottom: 4px';
titleDiv.innerHTML = '<code>Chem.Mol.__str__</code> call counter';
const countDiv = div.querySelector('div[class=widget-html-content]');
countDiv.style.textAlign = 'right';
div.insertBefore(titleDiv, label);
const TOP_PADDING = 120;
const RIGHT_PADDING = 100;
const viewPortWidth = window.innerWidth || document.body.clientWidth;
```


New PandasTools (using formatters)



The screenshot shows a JupyterLab environment with a notebook titled '1.3.2_new_pandastools.ipynb'. The notebook contains the following code:

```
[ ]: import_base64
import_pickle
import os
import pandas as pd
from io import BytesIO
import requests
from ipywidgets import widgets
from IPython.display import HTML, SVG, Image, Javascript, display
from threading import Thread
from datetime import datetime
import time

[ ]: pd.__version__

[ ]: from_rdkit import Chem, RDConfig
from_rdkit.Chem import rdMolDraw2D, IPythonConsole, MolsToGridImage

[ ]: os.getcwd()

[ ]: with open("JupyterDrawOptions.json") as hnd:
    rdMolDraw2D.UpdateMolDrawOptionsFromJSON(IPythonConsole.drawOptions, hnd.read())

[ ]: IPythonConsole.ipython_showProperties = True
IPythonConsole.ipython_useSVG = False

[ ]: from_rdkit.Chem import PandasTools

[ ]: bilastine_pubchem_url = "https://pubchem.ncbi.nlm.nih.gov/rest/pug/compound/name/bilastine/SDF"
```

The interface includes a top menu bar (File, Edit, View, Run, Kernel, Git, Tabs, Settings, Help), a toolbar with icons for file operations and execution, and a status bar at the bottom showing 'Simple' mode, 'Fully initialized', 'rdkit_master | Idle', 'Mem: 1.10 GB', 'Saving completed', 'Mode: Edit', 'Ln 11, Col 12', 'English (United States)', and '1.3.2_new_pandastools.ipynb'.



What next

Summary

- We have open-sourced on GitHub `rdkit-structure-renderer`, a JS app which enables interactive 2D molecule rendering in the browser
- The only dependency is on RDKit MinimalLib
- Interactive 2D molecule rendering can be enabled in Jupyter Lab with a single line of code and used for visualizing
 - Individual molecules
 - Grids of molecules with `MolsToGridImage`
 - pandas DataFrames through `PandasTools`
- We contributed improvements to `PandasTools` and scaffold alignment

Outlook

- Enable built-in magnifying lens functionality
- Enable further interactivity (molecule/atom property visualization, click/hover perception, etc.)
- Extend the interactivity to other environments (KNIME, .NET apps)
- Collect feedback from internal users and external community

Acknowledgments

- NIBR
 - Riccardo Vianello
 - Andreas Liistro
 - Nik Clare
 - Grégori Gerebtzoff
 - Robert Strulak
 - Daniil Tkachev
 - Ekaterina Stepanova
 - Jimmy Kromann
 - [...]
- Datagrok
 - Leonid Stolbov
 - Alex Paramanov
 - Andrew Skalkin
 - [...]
- RDKit
 - Greg Landrum
 - David Cosgrove
 - Brian Kelley
 - [...]
- Schrödinger
 - Nic Zonta
 - Rachel Walker
 - Ricardo Rodríguez
 - [...]



Thank you