

5_TYK2_Eliminating_PAINS

April 8, 2022

```
[1]: !git clone https://github.com/PatWalters/rd_filters ../tools
```

```
Cloning into '../tools'...
remote: Enumerating objects: 83, done.
remote: Total 83 (delta 0), reused 0 (delta 0), pack-reused 83
Unpacking objects: 100% (83/83), done.
```

```
[5]: !pip install docopt
```

```
Collecting docopt
  Downloading docopt-0.6.2.tar.gz (25 kB)
Building wheels for collected packages: docopt
  Building wheel for docopt (setup.py) ... done
  Created wheel for docopt: filename=docopt-0.6.2-py2.py3-none-any.whl
size=13705
sha256=d4a7ab80de8245e8b9d18fbfe0d8a39f0ff9c7629f5c7b4a6075515156f06dc3
  Stored in directory: /home/cv/.cache/pip/wheels/72/b0/3f/1d95f96ff986c7dffe46
ce2be4062f38ebd04b506c77c81b9
Successfully built docopt
Installing collected packages: docopt
Successfully installed docopt-0.6.2
```

```
[7]: !cd ../tools
```

```
[8]: !pip install ../tools
```

```
Processing /home/cv/TECHX/tools
Requirement already satisfied: pandas in
/home/cv/anaconda3/envs/deepchem/lib/python3.7/site-packages (from rd-
filters==0.1) (1.2.4)
Requirement already satisfied: docopt in
/home/cv/anaconda3/envs/deepchem/lib/python3.7/site-packages (from rd-
filters==0.1) (0.6.2)
Requirement already satisfied: python-dateutil>=2.7.3 in
/home/cv/anaconda3/envs/deepchem/lib/python3.7/site-packages (from pandas->rd-
filters==0.1) (2.8.1)
Requirement already satisfied: pytz>=2017.3 in
/home/cv/anaconda3/envs/deepchem/lib/python3.7/site-packages (from pandas->rd-
```

```

filters==0.1) (2021.1)
Requirement already satisfied: numpy>=1.16.5 in
/home/cv/anaconda3/envs/deepchem/lib/python3.7/site-packages (from pandas->rd-
filters==0.1) (1.20.2)
Requirement already satisfied: six>=1.5 in
/home/cv/anaconda3/envs/deepchem/lib/python3.7/site-packages (from python-
dateutil>=2.7.3->pandas->rd-filters==0.1) (1.15.0)
Building wheels for collected packages: rd-filters
  Building wheel for rd-filters (setup.py) ... done
  Created wheel for rd-filters: filename=rd_filters-0.1-py3-none-any.whl
size=33801
sha256=fe8bf2a4d3a9e8a02804fb53367885e106acd718b4c0b26c87f0a25d46658737
  Stored in directory: /tmp/pip-ephem-wheel-cache-
cfm_go60/wheels/de/3f/0f/9178deb02e296aabf3c32d4191e3bc9aad0e8bf9900044a50b
Successfully built rd-filters
Installing collected packages: rd-filters
Successfully installed rd-filters-0.1

```

```
[9]: !rd_filters -h
```

```

Usage:
rd_filters filter --in INPUT_FILE --prefix PREFIX [--rules RULES_FILE_NAME]
[--alerts ALERT_FILE_NAME] [--np NUM_CORES]
rd_filters template --out TEMPLATE_FILE [--rules RULES_FILE_NAME]

Options:
--in INPUT_FILE input file name
--prefix PREFIX prefix for output file names
--rules RULES_FILE_NAME name of the rules JSON file
--alerts ALERTS_FILE_NAME name of the structural alerts file
--np NUM_CORES the number of cpu cores to use (default is all)
--out TEMPLATE_FILE parameter template file name

```

```
[10]: !pwd
```

```
/home/cv/TECHX/notebooks
```

```
[13]: !rd_filters filter --in ../pipeline/5_PAINS_rd_filter/35_smiles.smi --prefix 22
      ↪N
      --rules ../tools/rd_filters/data/rules.json --alerts ../tools/rd_filters/data/
      ↪alert_collection.csv
```

```

using 8 cores
Using alerts from Inpharmatica
[21:27:49] Explicit valence for atom # 1 N, 4, is greater than permitted
[21:27:49] Explicit valence for atom # 15 N, 4, is greater than permitted
[21:27:49] Explicit valence for atom # 1 N, 4, is greater than permitted
[21:27:49] Explicit valence for atom # 19 N, 4, is greater than permitted

```

[21:27:49] Can't kekulize mol. Unkekulized atoms: 1 2 4 5 6 7 8

[21:27:49] Explicit valence for atom # 5 N, 4, is greater than permitted

[21:27:49] Can't kekulize mol. Unkekulized atoms: 16 17 18 20 22 23 24

Wrote SMILES for molecules passing filters to 22.smi

Wrote detailed data to 22.csv

22 of 35 passed filters 62.9%

Elapsed time 0.09 seconds

```
[20]: import pandas as pd

df = pd.read_csv('../pipeline/5_PAINS_rd_filter/22.smi', sep=' ', header=None)
df.columns = ['SMILES', 'Name']
```

```
[21]: df
```

```
[21]:
```

	SMILES	Name
0	<chem>O(c1ccc(cc1)[C@H]4n2nc(nc2NC(c3cc(OC)ccc3)C4)N)CC</chem>	ZINC13131776
1	<chem>Clc1c(ccc(Cl)c1)[C@H]3C[C@H](n2nc(nc2N3)N)c4cc...</chem>	ZINC13131298
2	<chem>O=C(c3cc2n1c(nnc1)c(nc2cc3)NC4CCCC4)c5ccccc5</chem>	ZINC06901595
3	<chem>s3c(SCc1nc(nc(n1)N)Nc2c(cccc2)C)nnc3NCc4ccccc4</chem>	ZINC09728763
4	<chem>Clc1c(Cl)ccc(c1)C3Nc2n(nc(n2)N)[C@@H](C3)c4ccc...</chem>	ZINC13131981
5	<chem>Fc1cc(ccc1)CNc3nc(c(c2onc(c2)C)cn3)c5cnc(N4CCO...</chem>	ZINC19124027
6	<chem>O=C(c3cc2n1c(nnc1C)c(nc2cc3)NC[C@H]4N(CCC4)CC...</chem>	ZINC20939758
7	<chem>O=C(c3cc2n1c(nnc1C)c(nc2cc3)NCc4ccncc4)c5ccccc5</chem>	ZINC20939765
8	<chem>O=C(N(C)C)c3ccc(c1nc(nc(n1)N)Nc2ccc(cc2)C)cc3</chem>	ZINC32540567
9	<chem>O(C(c3ccc(c1nc(nc(n1)N)Nc2ccc(cc2)C)cc3)C)C</chem>	ZINC32541042
10	<chem>Fc1c(c(F)ccc1)CNc4n2ncnc2nc(c3ccc(OC)cc3)c4</chem>	ZINC23432014
11	<chem>O(c1ccc(cc1)[C@H]4n2nc(nc2NC(c3ccc(OC)cc3)C4)N...</chem>	ZINC02472318
12	<chem>O(c1ccc(cc1)[C@H]4n2nc(nc2NC(c3cc(OC)ccc3)C4)N...</chem>	ZINC02472304
13	<chem>S=C(Nc2nc(c1c(O)cccc1)cc(n2)c3ccc(OC)cc3)N</chem>	ZINC01084111
14	<chem>Clc1ccc(cc1)C(=O)c3sc2n(c(=O)[nH]c(=O)c2c3N)c4...</chem>	ZINC02638281
15	<chem>Fc1ccc(cc1)C3Nc2n(nc(n2)N)[C@@H](C3)c5ccc(OCc4...</chem>	ZINC02472295
16	<chem>O(c1ccc(cc1)[C@H]4n2nc(nc2N[C@@H](c3ccc(cc3)C)...</chem>	ZINC02472298
17	<chem>[nH]2c1nc(NCCN(C)C)cc(c1cc2)c4cc3c(cnc(N)c3)cc4</chem>	ZINC72143513
18	<chem>s1cnc4c1cc(c3c2c([nH]cc2)nc(NCCN(C)C)c3)cc4</chem>	ZINC72146015
19	<chem>O(c1c(cccc1)CNc3nc4c(n2c3nnc2C)cc(cc4)C(=O)c5c...</chem>	ZINC09224703
20	<chem>o1c(nnc1)c5ccc(c4c2c([nH]cc2)nc(NCc3ccncc3)c4)cc5</chem>	ZINC72454514
21	<chem>n1(ncc4c1ccc(c3c2c([nH]cc2)nc(NCCN(C)C)c3)c4)C</chem>	ZINC72168947

```
[22]: !rd_filters filter --in ../pipeline/5_PAINS_rd_filter/all_smiles.smi --prefix_
      ↪all N
      --rules ../tools/rd_filters/data/rules.json --alerts ../tools/rd_filters/data/
      ↪alert_collection.csv
```

using 8 cores

Using alerts from Inpharmatica


```

[14:30:50] Explicit valence for atom # 20 N, 4, is greater than permitted
[14:30:50] Explicit valence for atom # 1 N, 4, is greater than permitted
[14:30:50] Explicit valence for atom # 22 N, 4, is greater than permitted
[14:30:50] Explicit valence for atom # 21 N, 4, is greater than permitted
[14:30:50] Explicit valence for atom # 1 N, 4, is greater than permitted
[14:30:50] Explicit valence for atom # 1 N, 4, is greater than permitted
[14:30:50] Explicit valence for atom # 17 N, 4, is greater than permitted
[14:30:50] Explicit valence for atom # 18 N, 4, is greater than permitted
[14:30:50] Explicit valence for atom # 1 N, 4, is greater than permitted
[14:30:50] Explicit valence for atom # 1 N, 4, is greater than permitted
[14:30:50] Explicit valence for atom # 12 N, 4, is greater than permitted
[14:30:50] Explicit valence for atom # 1 N, 4, is greater than permitted
[14:30:50] Explicit valence for atom # 1 N, 4, is greater than permitted
[14:30:50] Explicit valence for atom # 9 N, 4, is greater than permitted
[14:30:50] Explicit valence for atom # 9 N, 4, is greater than permitted
[14:30:50] Explicit valence for atom # 9 N, 4, is greater than permitted
[14:30:50] Explicit valence for atom # 1 N, 4, is greater than permitted
Wrote SMILES for molecules passing filters to all.smi
Wrote detailed data to all.csv
1826 of 2608 passed filters 70.0%
Elapsed time 0.82 seconds

```

```

[23]: df_rd_filter = pd.read_csv('../pipeline/5_PAINS_rd_filter/all.csv')
      df_rd_filter.head()

```

```

[23]:

```

	SMILES	NAME	FILTER	\
0	<chem>O(c1cc(ccc1)\C=C(\c2nc(nc(n2)N)Nc3ccccc3)/C#N)...</chem>	ZINC55039390	OK	
1	<chem>O(c1c(cccc1)\C=C\C(=O)Nc2nn4c(n2)N[C@@H](c3ccc...</chem>	ZINC09185435	OK	
2	<chem>O(c1ccc(cc1)[C@H]4n2nc(nc2N[C@@H](c3ccccc3)C4)...</chem>	ZINC02404424	OK	
3	<chem>O(c1ccc(cc1)[C@H]4n2nc(nc2N[C@@H](c3ccccc3)C4)...</chem>	ZINC05307450	OK	
4	<chem>Clc1c(cccc1)\C=C\C(=O)Nc2nn4c(n2)N[C@@H](c3ccc...</chem>	ZINC09158125	OK	

	MW	LogP	HBD	HBA	TPSA	Rot
0	388.431	3.28678	2	8	118.97	8.0
1	481.556	5.09350	2	7	90.30	7.0
2	439.519	4.61420	2	6	81.07	6.0
3	393.447	3.01790	2	7	90.30	6.0
4	485.975	5.73830	2	6	81.07	6.0

```

[28]: #identify which filters remove the largest number of molecules

```

```

from collections import Counter

def counter(df):
    count_list = list(Counter(df.FILTER).items())
    count_df = pd.DataFrame(count_list, columns=['Rule', 'Count'])
    count_df.sort_values('Count', inplace=True, ascending=False)

```

```

return count_df

counter(df_rd_filter)

```

```

[28]:

```

	Rule	Count
0	OK	2065
1	INVALID	399
4	Filter9_metal > 0	52
7	Filter82_pyridinium > 0	37
6	Filter20_hydrazine > 0	10
10	Filter14_thio_oxopyrylium_salt > 0	10
2	Filter39_imine > 0	9
9	Filter58_polyphenol2 > 0	7
3	Filter38_aldehyde > 0	4
11	Filter4_alpha_halo_carbonyl > 0	4
8	Filter94_2_halo_pyridine > 0	3
14	Filter44_michael_acceptor2 > 0	3
5	Filter78_bicyclic_Imide > 0	2
13	Filter81_Thiocarbamate > 0	2
12	Filter69_thio_carbonate > 0	1

```

[26]: len(df_rd_filter['NAME'].unique())

```

```

[26]: 1848

```

```

[27]: df_rd_filter_unique = df_rd_filter.drop_duplicates('NAME')
      df_rd_filter_unique.shape

```

```

[27]: (1848, 9)

```

```

[29]: counter(df_rd_filter_unique)

```

```

[29]:

```

	Rule	Count
0	OK	1457
1	INVALID	286
4	Filter9_metal > 0	34
7	Filter82_pyridinium > 0	30
10	Filter14_thio_oxopyrylium_salt > 0	9
6	Filter20_hydrazine > 0	7
2	Filter39_imine > 0	6
9	Filter58_polyphenol2 > 0	4
3	Filter38_aldehyde > 0	3
8	Filter94_2_halo_pyridine > 0	3
11	Filter4_alpha_halo_carbonyl > 0	3
13	Filter81_Thiocarbamate > 0	2
14	Filter44_michael_acceptor2 > 0	2
5	Filter78_bicyclic_Imide > 0	1

```
12      Filter69_thio_carbonate > 0      1
```

```
[ ]:
```