PAINS IN THE BUTT

John Mayfield

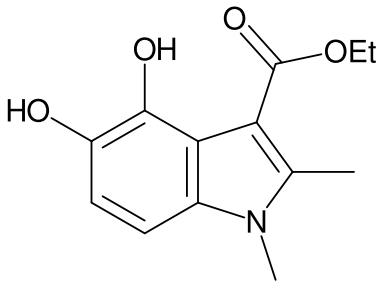
PAINS

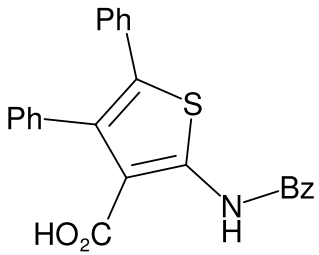
Pan Assay Interference Compounds (PAINS)

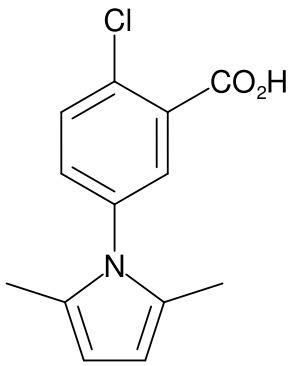
- High-Throughput Screening (HTS) frequent hitters
- Aggregation, Protein-reactive, Assay signalling interference

The PAINS filters are 480 **Sybyl Line Notation** (SLN) queries

- Six HTS campaigns using AlphaScreen over a screening library of 93K
- Substructure patterns used to **flag** false positives







JB Baell and GA Holloway. New Substructure Filters for Removal of Pan Assay Interference Compounds (PAINS) from Screening Libraries and for Their Exclusion in Bioassays. ***J. Med. Chem.*** 2010. 53, 2719–2740

Where it begins…

One of our tools raised a warning that one of the RDKit PAINS filters was **“always false”**.

Looking deeper I found more issues that (to my knowledge) have not been previously reported.

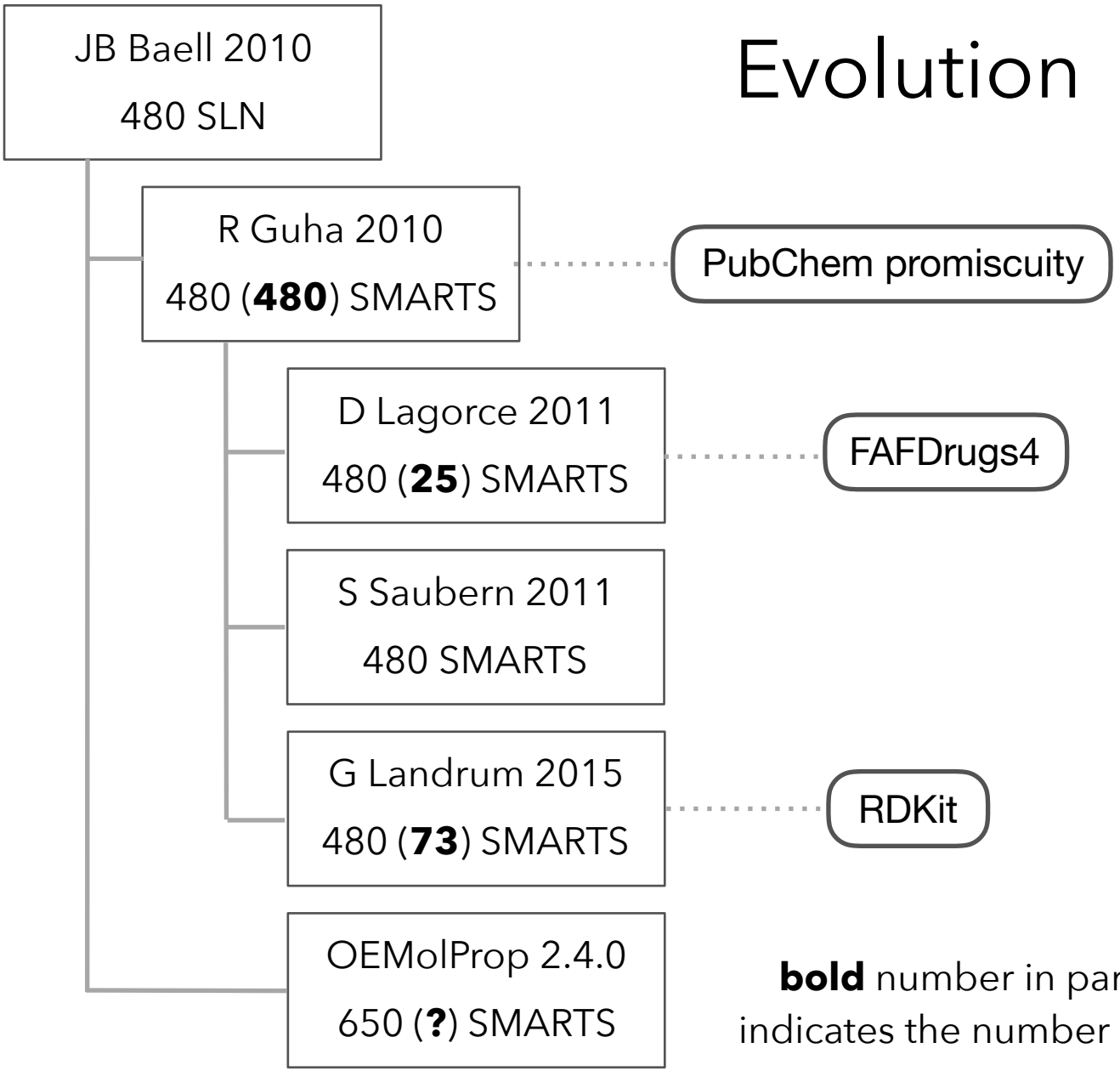
Disclaimer: this is **not** a talk about whether the filters are any good and how you should use them:

Peter Kenny’s blog: **http://fbdd-lit.blogspot.com/search/label/PAINS**

Capuzzi et al. Phantom PAINS: Problems with the Utility of Alerts for Pan−Assay INterference CompoundS. ***J. Chem. Inf. Model***. 2017, 57, 417−427

Baell and Nissink. Seven Year Itch: Pan-Assay Interference Compounds (PAINS) in 2017-Utility and Limitations. ***ACS Chem Biol.*** 2018 Jan 19;13(1):36-44

Evolution



bold number in parenthesis indicates the number of changes

Bibliography provided at end

SMARTS

Widely supported query language for matching substructures.

Powerful set of Atom/Bond expressions

SMARTS Atomic Primitives

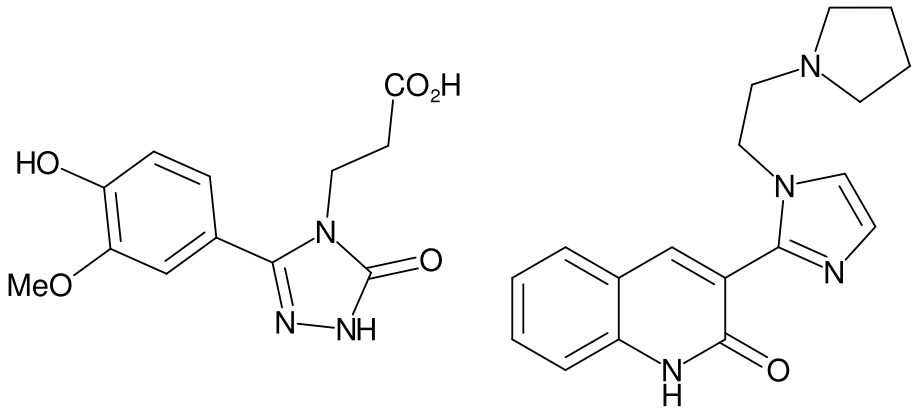
Symbol	Symbol name	Atomic property requirements	Default
*	wildcard	any atom	(no default)
a	aromatic	aromatic	(no default)
A	aliphatic	aliphatic	(no default)
D<n>	degree	<n> explicit connections	exactly one
H<n>	total-H-count	<n> attached hydrogens	exactly one[1]
h<n>	implicit-H-count	<n> implicit hydrogens	at least one
R<n>	ring membership	in <n> SSSR rings	any ring atom
r<n>	ring size	in smallest SSSR ring of size <n>	any ring atom[2]
v<n>	valence	total bond order <n>	exactly one[2]
X<n>	connectivity	<n> total connections	exactly one[2]
x<n>	ring connectivity	<n> total ring connections	at least one[2]
- <n>	negative charge	-<n> charge	-1 charge (-- is -2, etc)
+<n>	positive charge	+<n> formal charge	+1 charge (++ is +2, etc)
#n	atomic number	atomic number <n>	(no default)[2]

http://www.daylight.com/dayhtml/doc/theory/theory.smarts.html

SLN vs SMARTS differences

SLN semantics do not translate exactly to SMARTS leading to false positives/negatives.

Baell reports **hzone_phenol_B** and **dyes5a** false positives when using SMARTS:

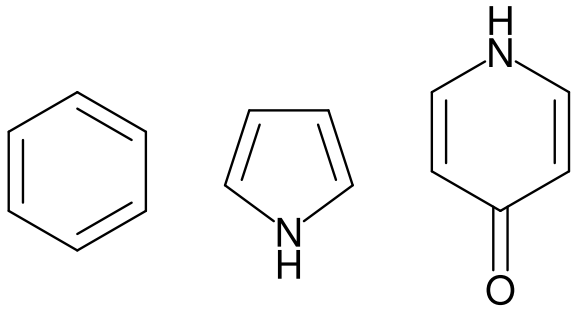


…actually **false negative** due to aromatic carbonyl but implementation problem nevertheless

Baell and Nissink. Seven Year Itch: Pan-Assay Interference Compounds (PAINS) in 2017-Utility and Limitations. ***ACS Chem Biol.*** 2018 Jan 19;13(1):36-44

Variable 1: Aromaticity

There are different models of aromaticity:



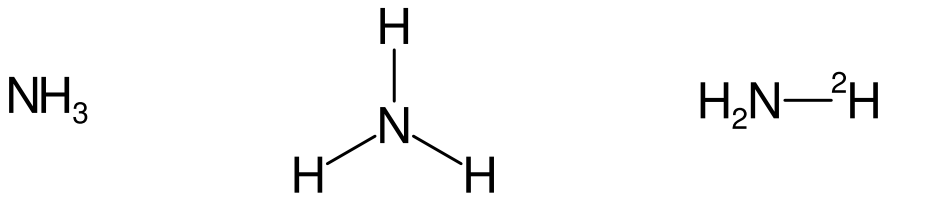
MDL-like	☑	☐	☐
Tripos-like	☑	☑	☐
Daylight-like	☑	☑	☑

Use correct model for a method (e.g. ALogP, ECFP, MMFF, etc). OEChem, CDK, and RDKit allow user to choose.

PAINS SMARTS can be run against a Tripos-aromatic molecules (see R Guha 2010)!

Variable 2: Hydrogen Representation

Hydrogens can be represented **implicitly** as a count on an atom or **explicitly** as atoms in the connection table:



Implicit | Explicit | Implicit+Explicit

Terminal hydrogens without a charge, isotope, or atom-map can be safely converted between implicit and explicit.

We can run a SMARTS against a molecule with implicit (preferred) or explicit hydrogens.

Roger Sayle. Explicit and Implicit Hydrogens: Taking liberties with valence
https://nextmovesoftware.com/blog/2013/02/27/explicit-and-implicit-hydrogens-taking-liberties-with-valence/

Writing queries…

Writing good SMARTS is tricky, how would **you** match a Methyl?

- `*C` will match a lot more than just methyl
- `*[CH3]` better, but still might match things we don't want
- `*-[0CD1v4!Rh3+0]` best
- `*-[0CX4v4!RH3+0]` bester (allows explicit H but also ^{2}H!)

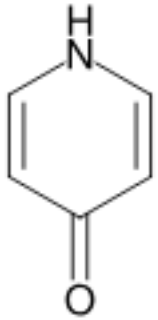
Roger's fundamental SMARTS axiom:

X = D + h

Noel O'Boyle. SMARTS for Dummies: *https://baoilleach.blogspot.com/2018/11/smarts-for-dummies.html*

SMARTS Aromaticity

A query can be made agnostic to aromaticity model:




```
O=c1ccncc1 a
O=c:1:c:c:n:c:c:1 b
O=C1C=CNC=C1 c
O=C-1-C=C-N-C=C-1 d
O=[#6]1[#6]=,:[#6][#7][#6]=,:[#6]1 e
```

	PubChem Compound (96M)	
	Daylight Aromaticity	**Tripos Aromaticity**
a	270623	0
b	269292	0
c	126	89343
d	126	89343
e	269853	269268

SMARTS Hydrogen Suppression 1

The SMARTS **`C([H])[H]`** (or **`C([#1])[#1]`**) can be made to match both implicit and explicit hydrogens:

`[CH2]`	**Wrong!**
`[C!H0!H1]`	
`[CH{2-4}]`	CACTVS, RDKit, and CDK extension
`[CH>1]`	NextMove extension

Can safely be automated by software.

The RDKit has **MergeQueryHs** but an "or" was mistakenly used instead of an "and" in the **thiophene_E** pattern:

`[#6;!H0,!H1]` *wrongly simplifies to* **`[#6]`**

SMARTS Hydrogen Suppression 2

Suppressing optional hydrogens is more complex, for example:

`c([#1,CH3])([#1,CH3])`

We need add the three possibilities:

`!H0!H1`	H and H
`[!H0][CH3]`	H and Me (also Me and H)
`*([CH3])([CH3])`	Me and Me

`[c;!H0!H1,$([!H0][CH3]),$(*([CH3])([CH3]))]`

Possible to automate… **care must be taken!**

RDKit and FAFDrugs changes

RDKit Aromaticity Changes

anil_NH_alk_A
anil_di_alk_coum
anthranil_acid_J
colchicine_B
coumarin_A,B,C,D,E,F,G,H
cyano_pyridone_A,B,D,E,F,G
ene_five_het_J
ene_rhod_H
ene_six_het_B
het_55_A
het_65_B,E,L
het_666_C
het_thio_5_C **(+)**
het_thio_N_5A,65A **(+)**
thiaz_ene_B **(+)**
thio_carbonate_A,B **(+,)**
thio_ester_A,C
thio_urea_C,H,P
anil_di_alk_dhp
dhp_amidine_A
naphth_amino_A,C,D

RDKIt Hydrogen Changes

anil_OC_alk_C
anil_di_alk_A,C,E,F
anisol_A
dhp_keto_A
ene_quin_methide
ene_rhod_C,D
het_5_pyrazole_OH
het_pyridiniums_A,B **(+,+)**
het_thio_666_A
het_thio_676_A
hzone_acyl_misc_A
hzone_enamin **(+)**
hzone_furan_A,B
hzone_phenone
hzone_thiophene_A,B
indol_3yl_alk
melamine_A
styrene_B
thiaz_ene_E
thiophene_E
thiophene_amino_Ab

RDKit Mixed Changes

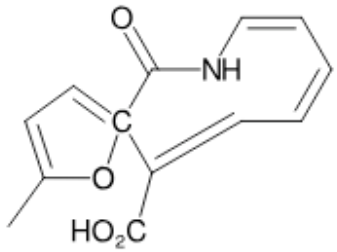
anthranil_one_A
het_65_pyridone_A
thiaz_ene_A

FAFDrugs2 Changes

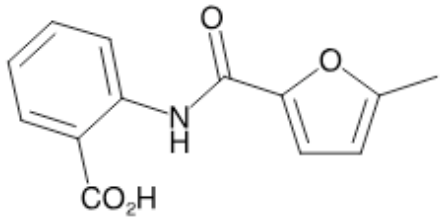
acyl_het_A
anil_alk_thio
anisol_B
anthranil_acid_D
aphtha_amino_C
diazox_sulfon_A
dyes3A
ene_five_het_B
ene_one_hal
ene_rhod_J
hzone_anil_di_alk
hzone_phenol_A
imidazole_B
quinone_A
thiaz_ene_C
thio_dibenzo
thio_ketone
thiophene_amino_B

Issue 1: Ring Closures

The SMARTS initially flagged by our tool was **anthranil_acid_I**.
It was a mistake in the SLN to SMARTS translation:

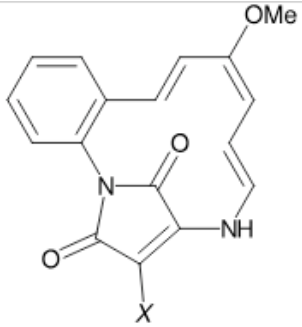


before

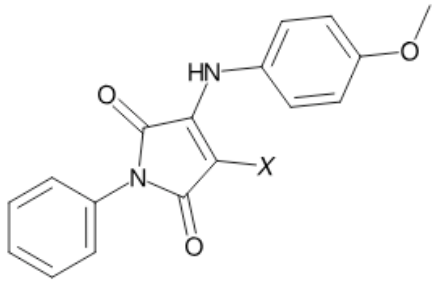


after

Further analysis identified **anil_OC_alk_B** as the same



before

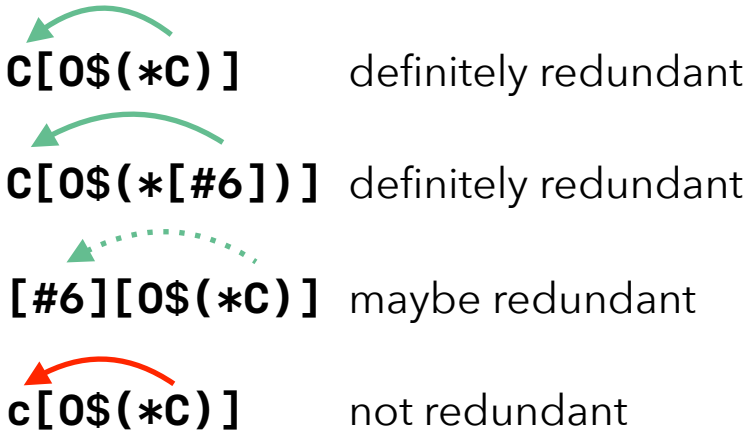


after

Both FAFDrugs and RDKit missed this!

Issue 2: Recursive Definitions

Matching optional terminal groups uses recursive SMARTS.
Recursive SMARTS can match back onto the parent:



CACTVS adds the **`$$(..)`** extension that does not allow the subquery to match on the atoms already matched by the parent.

Issue 2: Recursive Definitions

Writing an algorithm to check this identified 8 patterns:

Definitely redundant

het_thio_666_A

hzone_enamin

anil_di_alk_D

Maybe Redundant

het_thio_676_A

dhp_keto_A

thiophene_E

thiaz_ene_A

anil_di_alk_E

het_thio_666_A

```
..C:C(~Any[IS=H,NHC[TAC=4],C:C]).. SLN
..cc~[$([#1]),$([#7](-[#1])-[#6;X4]),$([#6]:[#6])].. R Guha
..cc[c;!H0,$(c~[#7](-[#1])-[#6;X4]),$(c~[#6]:[#6])].. RDKit
```

to fix we can multiply out the possibilities

```
..cc[c;!H0]..
..ccc(~[#7](-[#1])-[#6;X4])..
..ccc(~[#6]:[#6])..

or

..cc[c;!H0,$(c~[#7](-[#1])-[#6;X4]),$(c(cc)~[#6]:[#6])]..
```

OEChem has 650 patterns: *"in certain cases it isn't possible to exactly represent the SLN queries given in the paper with a single SMARTS, so in those cases the rules have been **split out into multiple SMARTS** patterns"* - OEMolProp documentation

Known Unknown Unknowns?

Previous publications have validated the SLN vs SMARTS on a **10K** random compounds (selected from the original 93K).

320/480 of the reference SLN had **0** hits in the **10K**

If the SLN didn't match and the SMARTS didn't match we can't tell there was any difference.

[Sn] and **[Pb]** both have zero hits in ChEMBL 24, but obviously not the same pattern!

D Lagorce *et al.* (2011) the full list of hits in supplementary section 8.

S Saubern *et al.* (2011) supplementary Table A (45 queries found) and B (115 queries not found)

More Power!

Running SMARTS over large structure sets can be **slow** making it time consuming to curate large sets of SMARTS.

For reference *Hilbig and Rarey* (2015) report **42h26m** to check **1.172M** compounds for the PAINS patterns.

With the right tools we can check a datasets **~100x** larger in **32s**

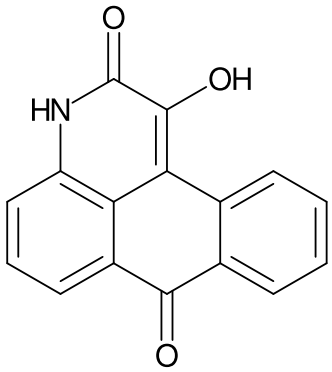
```
$ time ./atdbgrep -j 16 wehi_pains_rdkit.sma pubchem.smi.atdb -s
pubchem.smi > pubchem_pains.smi

real  0m31.228s
user  5m54.840s
sys   0m11.727s
```

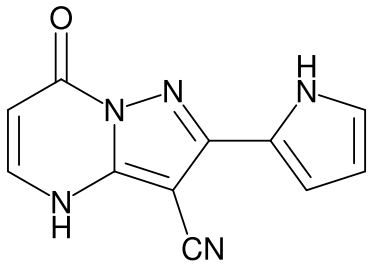
Matthias Hilbig and Matthias Rarey. MONA 2: A Light Cheminformatics Platform for Interactive Compound Library Processing. ***J. Chem. Inf. Model.***, 2015, 55 (10)

Issue 3: Aromaticity

After making changes **25/480** filters still hit nothing in PubChem Compound (Oct 2018)!

8 patterns could be updated to account for aromaticity (i.e. missed in Greg's initial curation).






```
thiaz_ene_C, het_6_imidate_A, quinone_C, thio_urea_K,
het_6_imidate_B, colchicine_het, het_65_H, het_5_inium
```

Change in hit count

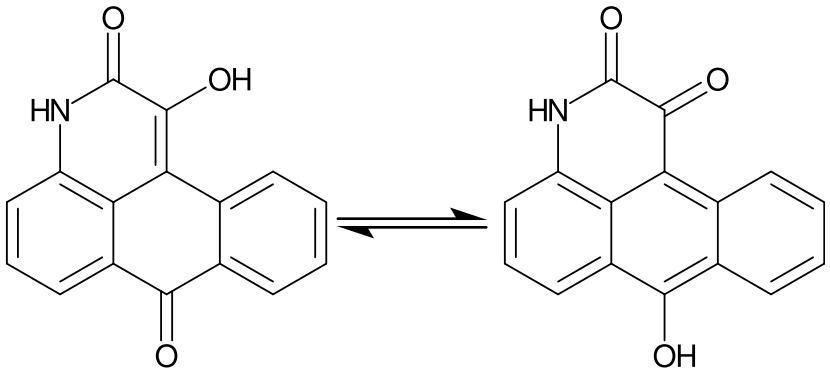
	eMolecules Jan 2019		PubChem Compound Oct 2018	
	Before	**After**	**Before**	**After**
thiaz_ene_C	0	193	0	413
quinone_C	0	6	0	0
thio_urea_K	0	3	0	0
het_5_inium	0	3	0	0
het_thio_666_A	877	569	11139	4721
dhp_keto_A	286	286	2931	2931
thiaz_ene_A	0	7189	80	21071
thiophene_E	0	0	1138	1144
hzone_enamin	232	232	3256	3063
anil_di_alk_D	46600	20393	247897	136436
anil_di_alk_E	28538	28538	194575	194575
anil_OC_alk_B	0	982	0	1099
anthranil_acid_I	0	8	0	26

… other changes not shown

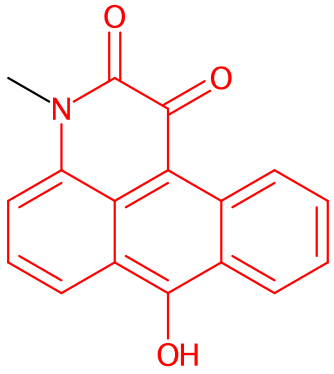
Issue 4: Tautomers

Remaining **no hitters** identify issues with tautomers

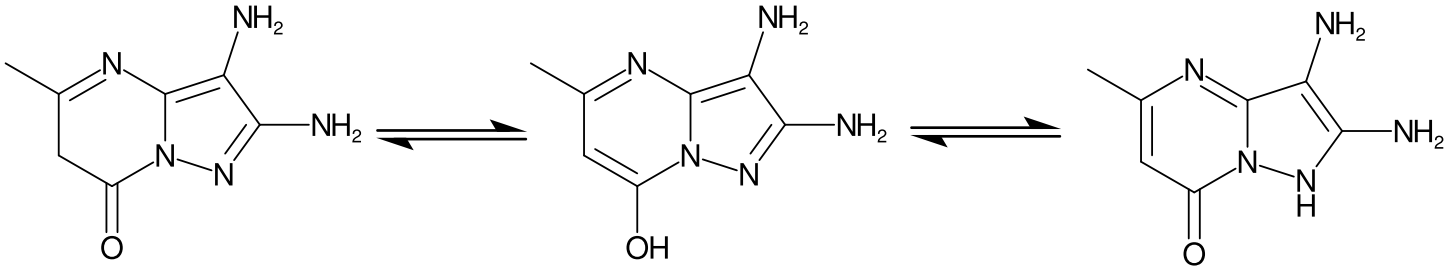
- also a problem with the **original SLN!**



quinone_C filter



CID 5409668

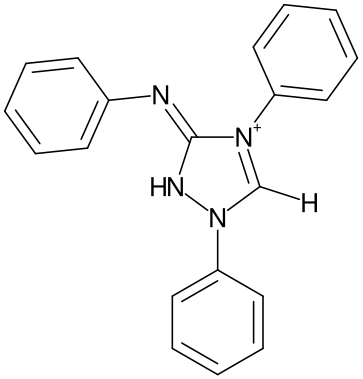


het_65_G filter

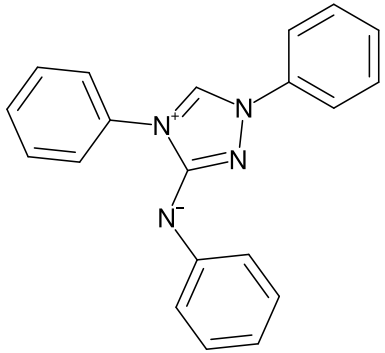
CID 4060544

Issue 4: Resonance

Moving target…

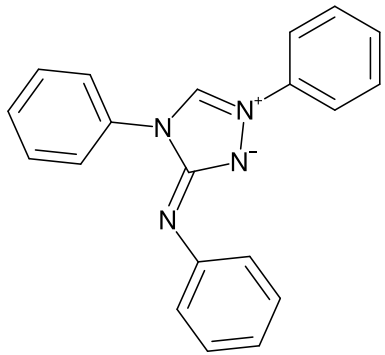


het_5_inium filter



CID 720071

(Oct 2018)



CID 720071

(Feb 2019)

Of the 480 filters it's possible a large fraction are tautomer specific. We could **fix the patterns** or **improve query tools**?

Conclusions

- Writing SMARTS is hard
- Potential problems in SMARTS can be automatically detected or avoided
- Excellent curation work done by Greg in 2015!
  - Changes to SMARTS have been submitted as a patch
  - RDKit "recently" added an **MDL-like model** of aromaticity, it might be useful to add a **Tripos-like model**
- Fast substructure search (i.e. **Arthor**) enables fast curation.

- Future Work
  - Tautomer independent substructure search
  - More PAINS curation, tautomers and FAFDrugs changes

Thanks

Greg Landrum

Rajarshi Guha

Noel O'Boyle and Roger Sayle

Willem Nissink

Wolf Ihlenfeldt

Marc Nicklaus (suggested the title)

We're hiring!

info@nextmovesoftware.com

If you found this presentation interesting you might be qualified for a job at NextMove Software!

Bibliography

JB Baell and GA Holloway. New Substructure Filters for Removal of Pan Assay Interference Compounds (PAINS) from Screening Libraries and for Their Exclusion in Bioassays. ***J. Med. Chem.*** 2010. 53, 2719–2740

R Guha. PAINS Substructure Filters as SMARTS. 2010. http://blog.rguha.net/?p=850

G Landrum. Curating the PAINS filters. 2015. http://rdkit.blogspot.com/2015/08/curating-pains-filters.html

S Saubern, R Guha, and JB Baell. KNIME Workflow to Assess PAINS Filters in SMARTS Format. Comparison of RDKit and Indigo Cheminformatics Libraries. ***Mol. Inf.*** 2011, 30, 847–850

D Lagorce, J Maupetit, JB Baell, O Sperandio, P Tufféry, MA Miteva, H Galons, BO Villoutreix. The FAF-Drugs2 server: a multistep engine to prepare electronic chemical compound collections. Bioinformatics. 2011, 27(14), 2018-20

SA Canny, Y Cruz, MR. Southern and PR Griffin. PubChem promiscuity: a web resource for gathering compound promiscuity data from PubChem. ***Bioinformatics***. 2012, 28(1), 140-141

JB Baell and W Nissink. Seven Year Itch: Pan-Assay Interference Compounds (PAINS) in 2017-Utility and Limitations. ***ACS Chem Biol.*** 2018 Jan 19;13(1):36-44