258th ACS National Meeting San Diego Aug 2019

Making a hash of it

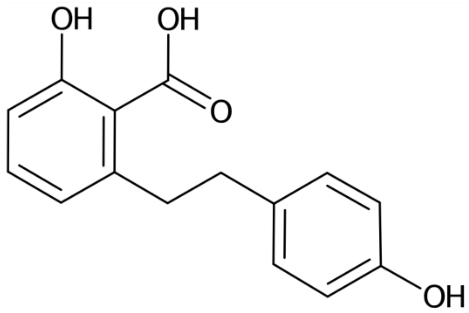
The advantage of selectively leaving out structural information

Noel O’Boyle and Roger Sayle

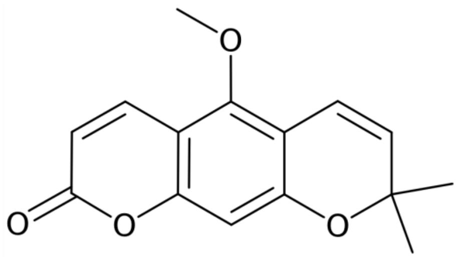
NextMove Software

INTRODUCTION

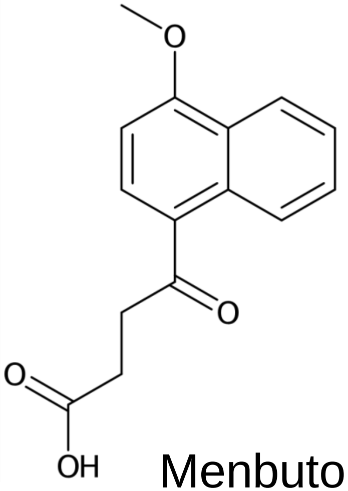
WHAT DO THESE MOLECULES HAVE IN COMMON?



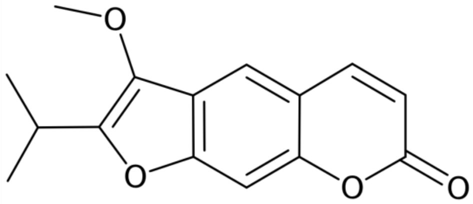
Lunularic acid



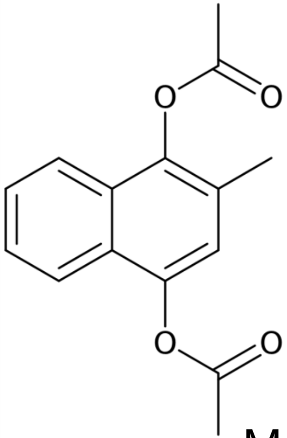
Xanthoxyletin



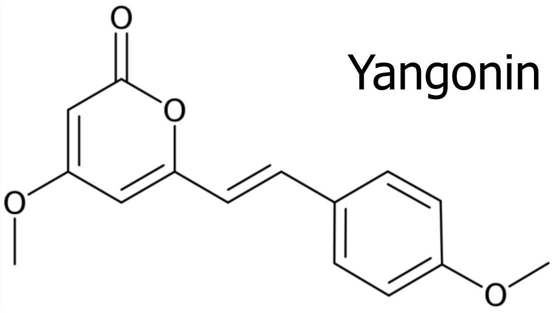
Menbutone



Peucedanin

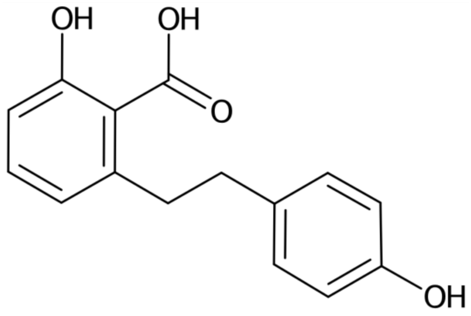


Menadiol diacetate

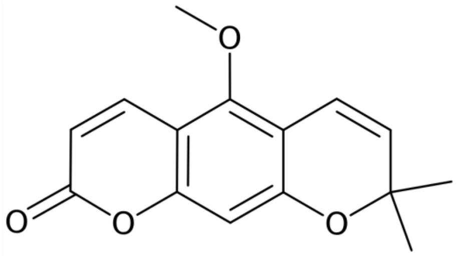


Yangonin

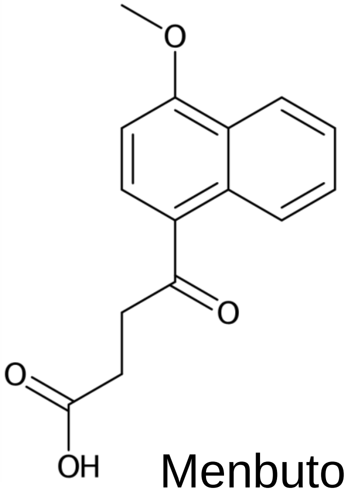
WHAT DO THESE MOLECULES HAVE IN COMMON?



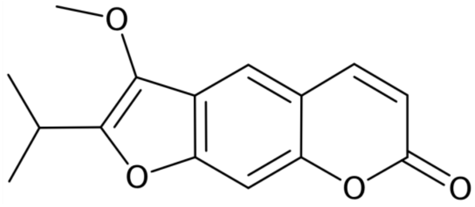
Lunularic acid



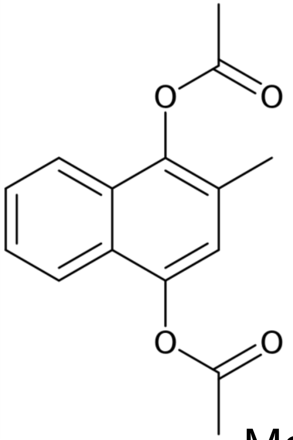
Xanthoxyletin



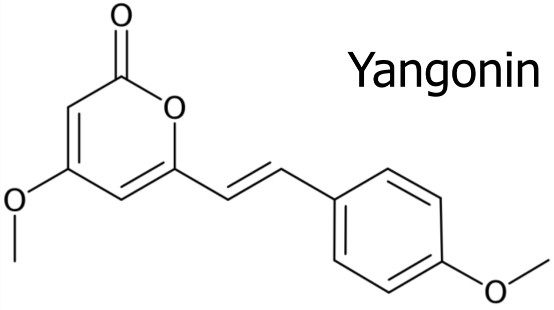
Menbutone



Peucedanin



Menadiol diacetate



Yangonin

Listed together in the Merck Index, 13th Edition, under **$C_{15}H_{14}O_4$.**

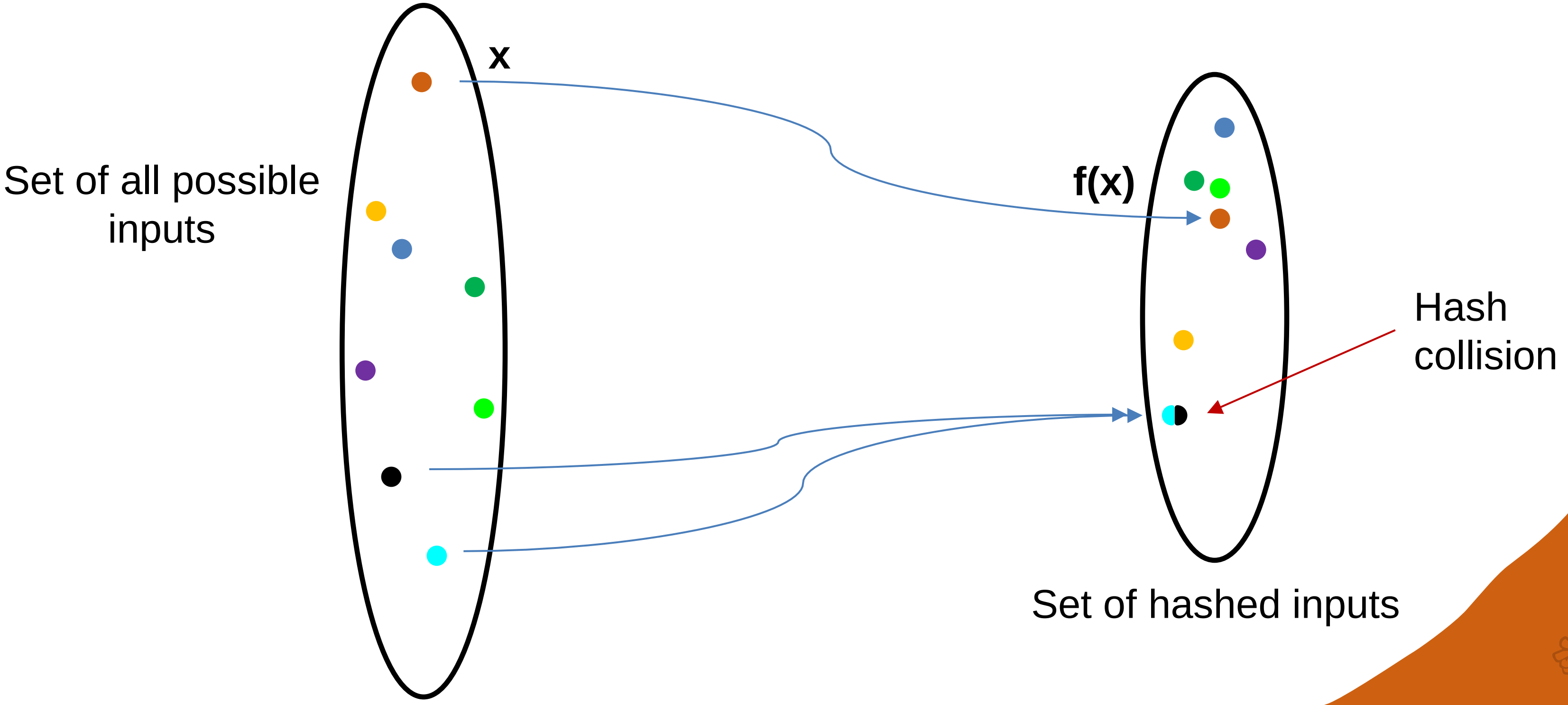
MOLECULAR FORMULA

- A representation of the molecular structure that discards connectivity and stereochemistry
  - Sufficient to calculate the molecular weight
  - The simplest molecular hash

- Can be used to index chemical structures
  - Invariant to differences in bond representation (e.g. inorganic complexes), tautomeric form
- Can be used to identify isomers
  - Constitutional isomers and stereoisomers

HASH FUNCTION

- A function (or procedure) that maps a set of inputs to a smaller set of outputs
  - Inputs with the same hashed value ("hash") are equivalent (in some respect)
- Can be used to find/eliminate duplicates (with respect to some property)



HASHING WORDS VS MOLECULES

FIND ALL ANAGRAMS IN A SET: ENUMERATION

- For all pairs of words (A, B) of the same length in the set, generate all possible orderings of the letters of A and compare to B

- Drawbacks:
  - The number of pairs of words increases with ~N^2
  - Enumerating all possible orders of a word of length N may take a while
    - E.g. 13! is 62270208

- Is there a more efficient way?
  - (Hint: enumeration is rarely a good idea)

FIND ALL ANAGRAMS IN A LIST OF WORDS

- For each word, create a hash of the letters in sorted order, for example:
  - “noboyle” → `belnooy`
  - “boloney” → `belnooy`
- Collate words that hash to the same value
  - `belnooy`: [noboyle, boloney]
- Efficient process, future queries just need a lookup
- General points
  - We need a hash that is only shared by the items of interest
    - **All** anagrams of “noboyle” have the hash `belnooy`
    - **Only** anagrams of “noboyle” have the hash `belnooy`
  - Hashes discard some features of the original data
    - Here we discard information on the original letter order

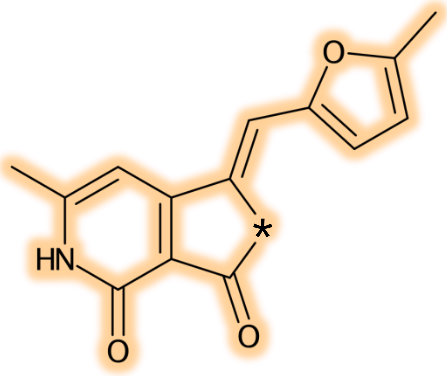
Find anagrams	Find duplicate molecules
Set of words	Set of molecules (e.g. SMILES strings)
Sort letters in alphabetical order	Sort atoms in canonical order
Generate hash (sorted text)	Generate hash (canonical SMILES)
Find duplicates based on hash	Find duplicates based on hash
Original words are anagrams of each other	Original SMILES represent the same molecule

FIND WORDS THAT DIFFER BY A SINGLE LETTER

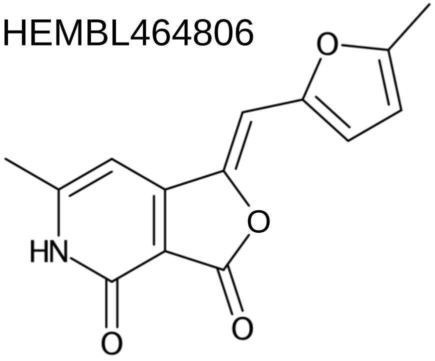
- For each word, create hashes where each letter in turn is replaced by an asterisk

- isotropic
  - `*sotropic` (also **e**sotropic)
  - `i*otropic` (also i**n**otropic)
  - `is*tropic`
  - `iso*ropic` (also iso**r**ropic)
  - `isot*opic`
  - `isotr*pic`
  - `isotro*ic` (also isotro**n**ic)
  - `isotrop*c`
  - `isotropi*` (also isotropi**l**)

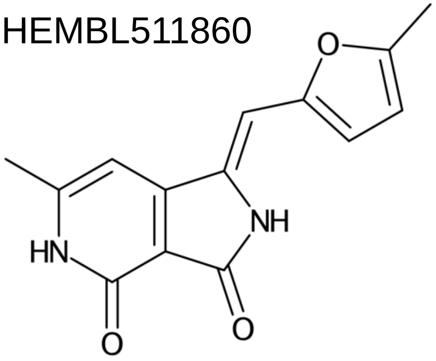
FIND MOLECULES THAT DIFFER BY A SINGLE ATOM



CHEMBL464806



CHEMBL511860

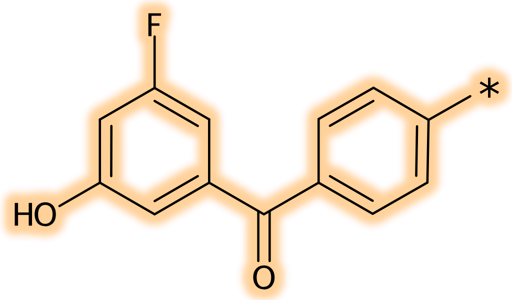


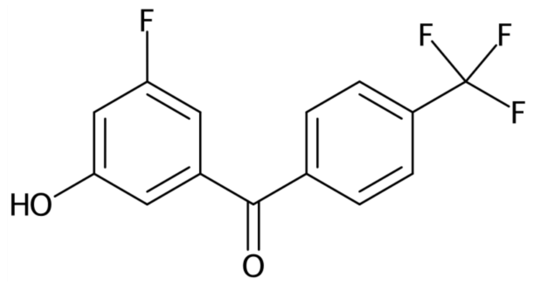
`Cc1cc2c(c(=O)[nH]1)C(=O)*/C2=C\c1ccc(C)o1`

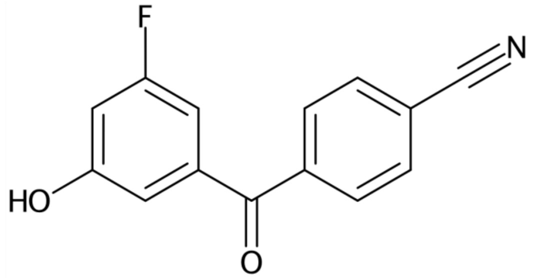
- For each atom in turn, create a hash as follows:
  - 1. Set to zero the atomic number, charge, isotope, and implicit hydrogen count
  - 2. Generate the canonical SMILES
- Molecules with the same hash differ by single atom replacement

Words that differ by a single letter	**Molecules that differ by a single atom**
Set of words	Set of molecules (e.g. SMILES strings)
Replace each letter in turn with an asterisk	For each atom in turn, set to zero the atomic number, charge, isotope and implicit hydrogen count
Generate hash (text after replacement)	Generate hash (canonical SMILES)
Find duplicates based on hash	Find duplicates based on hash
Original words have one letter replaced	Original molecules have one atom replaced

IDENTIFY MATCHED PAIRS IN A SET







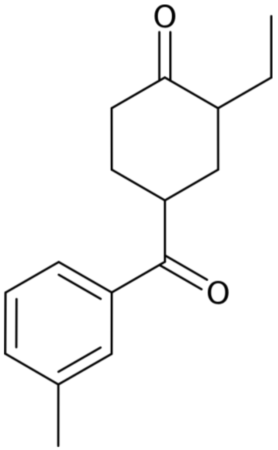
`Fc1cc(O)cc(c1)C(=O)c1ccc(*)cc1`

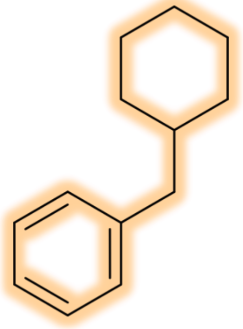
- Replace each R group in turn by an asterisk, and generate the canonical SMILES
- Matched pairs share the same hash
- References:
  - Wagener and Lommerse (*JCIM*, **2006**, *36*, 677) used Cactvs hash codes
  - Popularised by Hussain and Rea (*JCIM*, **2010**, *50*, 339) using canonical SMILES
  - Our implementation: *J. Med. Chem.*, **2014**, *57*, 2704

MOLHASH

https://github.com/nextmovesoftware/molhash

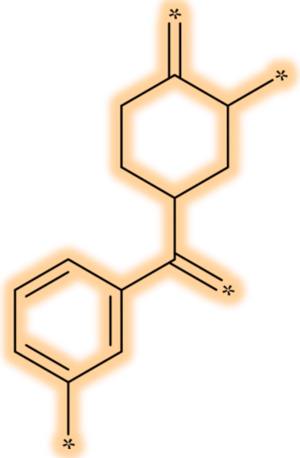
Murcko scaffold hashes





c1ccc(CC2CCCCC2)cc1

Murcko scaffold hash

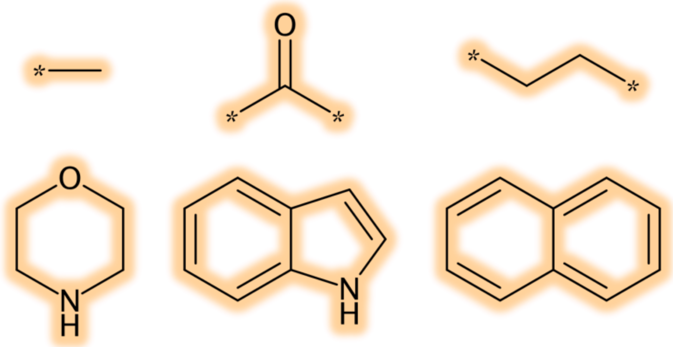


c1cccc(C(=)C2CCC(=*)C(*)C2)c1

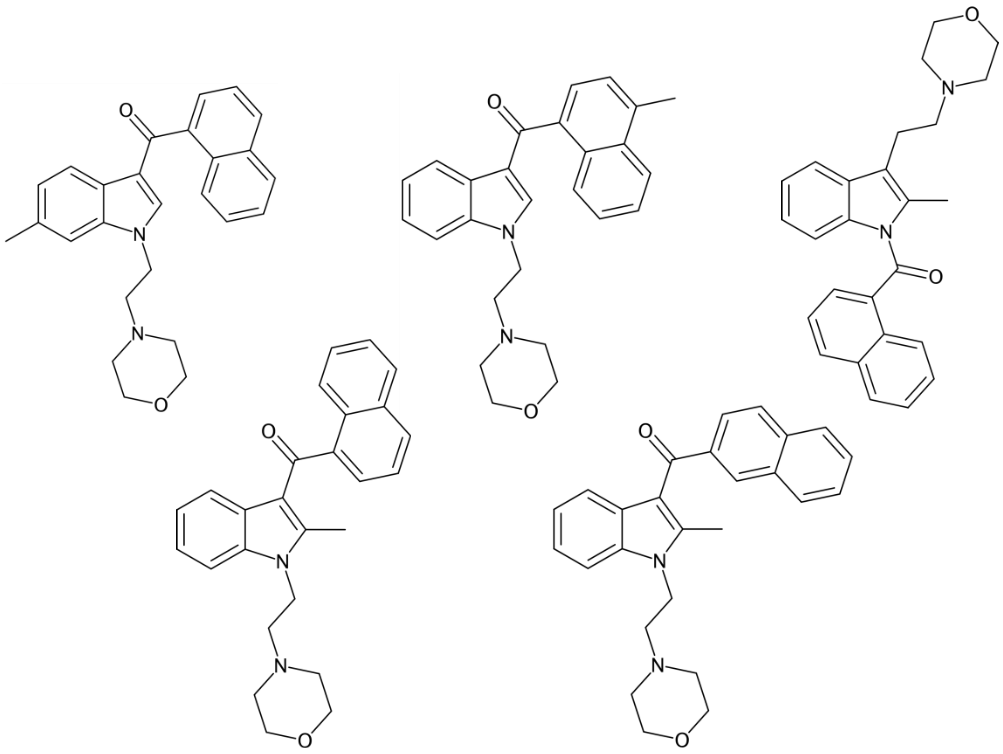
Extended Murcko scaffold hash

- The Murcko scaffold hash is the canonical SMILES after removing all substituents
- The extended Murcko scaffold hash is the canonical SMILES after replacing all substituents with attachment points

REGIOISOMER HASH

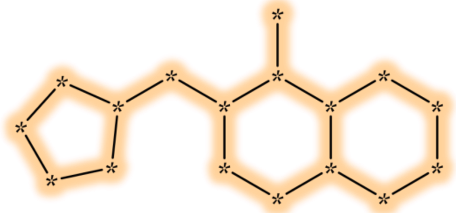


```
*C.*C(*)=O.*CC*.C1COCCN1.c1c
cc2[nH]ccc2c1.c1ccc2ccccc2c1
```

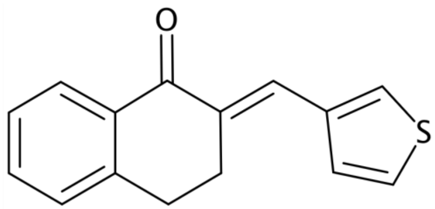


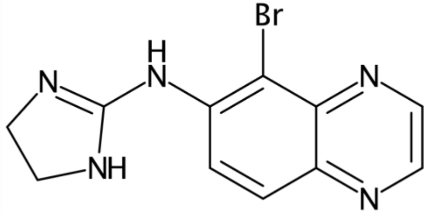
- The canonical SMILES after breaking a subset of acyclic single bonds and replacing the connection by an asterisk or hydrogen.
  - Specifically, acyclic single bonds are cut if either end of the bond is involved in a ring or if the bond is between a non sp^2-hybridized carbon atom and a non-carbon atom

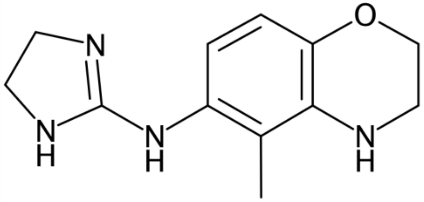
ANONYMOUS GRAPH

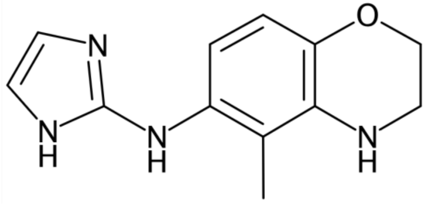


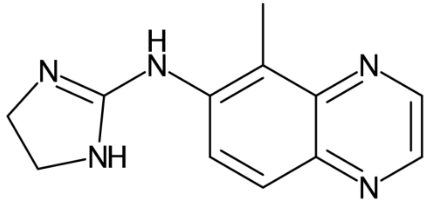
1*(2****2)***2*****21

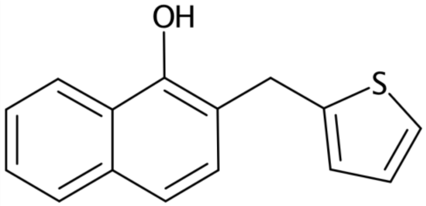


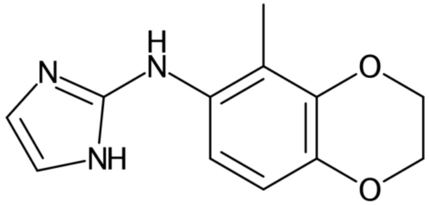






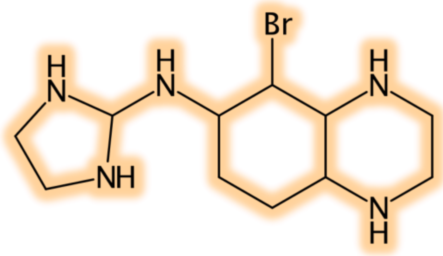




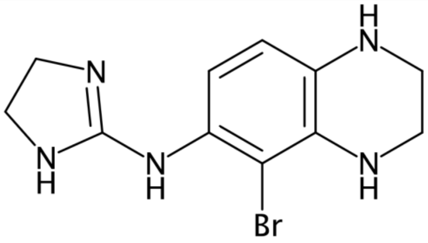


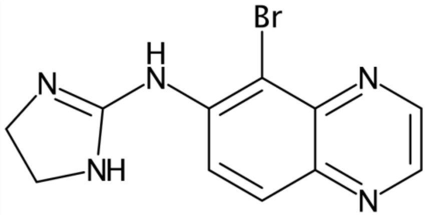
- The canonical SMILES string after setting all atoms to asterisks and bonds to single bonds
- Can identify molecules that share the same graph structure independent of atom identity, bond order or hydrogen count

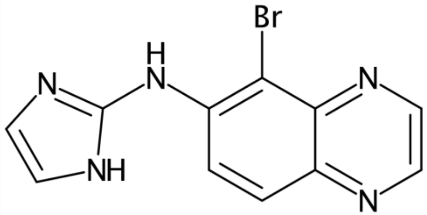
ELEMENT GRAPH



BrC1C(NC2NCCN2)CCC2NCCNC21

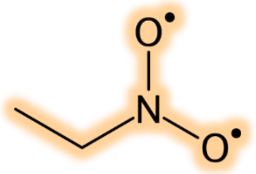




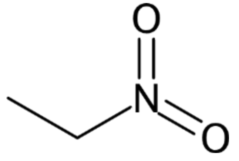


- The canonical SMILES after setting all bonds to single bonds, and normalizing hydrogen counts
- Identify molecules that share the same bonding arrangement but different bond order

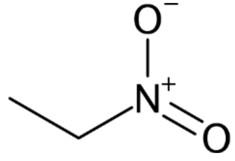
MESOMER HASH



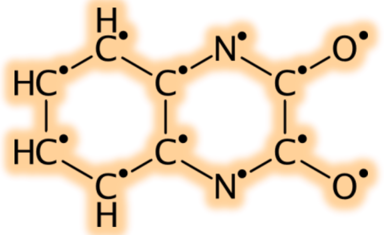
CCN([O])[O]...



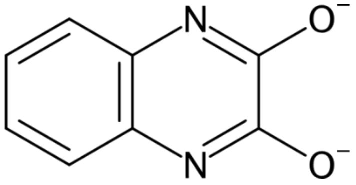
..._0



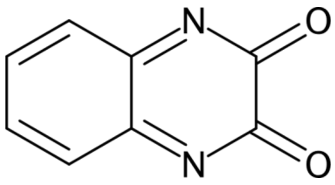
..._0



[CH]1[CH][CH][C]2[C]([CH]1)[N][C]
([C]([N]2)[O])[O]...



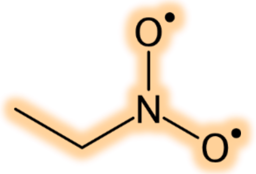
..._-2



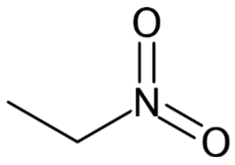
..._0

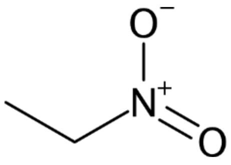
- Two parts: a SMILES component and an integer
  - The canonical SMILES after setting all bonds to single bonds, all charges to zero
  - An integer: the total charge
- Identify duplicate structures independent of resonance form

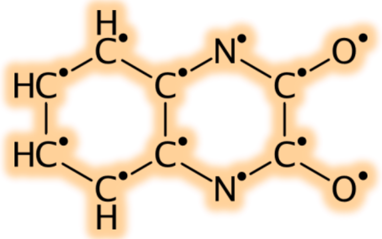
REDOX PAIR HASH



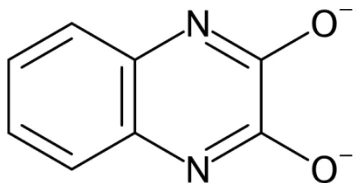
CCN([O])[O]

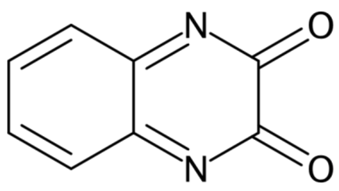






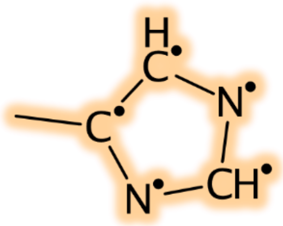
[CH]1[CH][CH][C]2[C]([CH]1)[N][C]([C]([N]2)[O])[O]



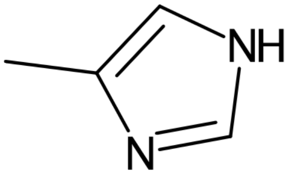


- ~~Two~~One part: a SMILES component ~~and an integer~~
  - The canonical SMILES after setting all bonds to single bonds, all charges to zero
  - ~~An integer: the total charge~~
- Identify redox pairs (same hash, different charge)

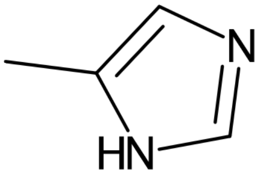
HETEROATOM TAUTOMER HASH



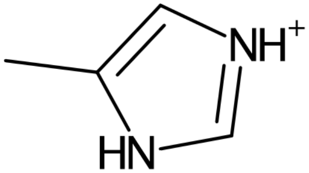
C[C]1[CH][N][CH][N]1...



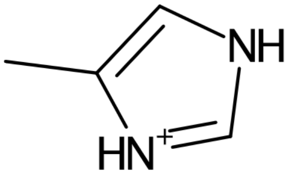
...1_0



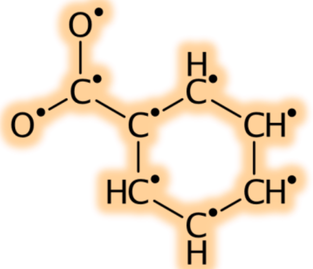
...1_0



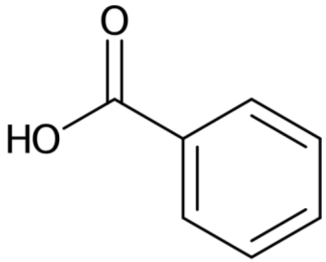
...2_1



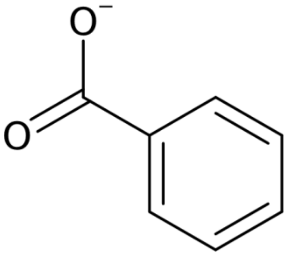
...2_1



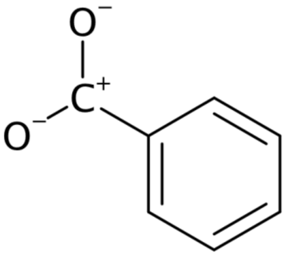
C[C]1[CH][N][CH][N]1...



...1_0



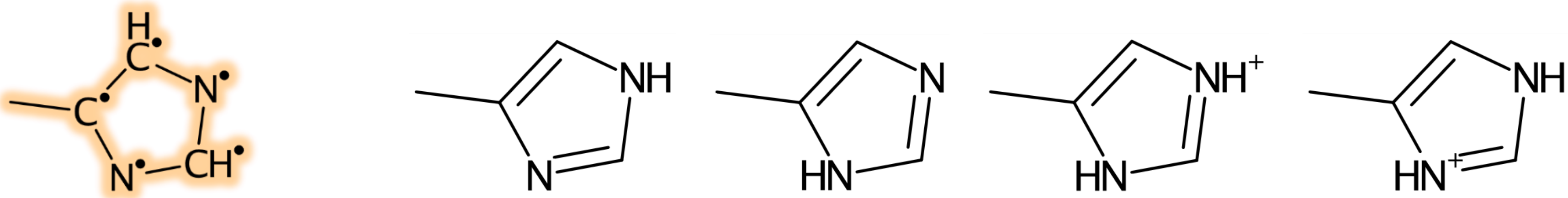
...0_-1



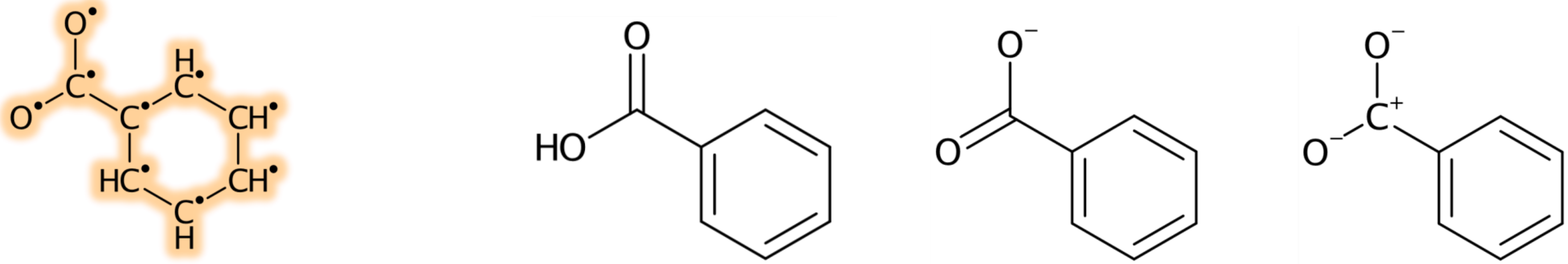
...0_-1

- Two parts: a SMILES component and two integers
  - The canonical SMILES after setting all bonds to single bonds, all charges to zero, and stripping hydrogens from heteroatoms
  - Two integers: the count of stripped hydrogens, and total charge
- Identify duplicate structures independent of tautomeric state or resonance form

HETEROATOM PROTOMER HASH



C[C]1[CH][N][CH][N]1_1

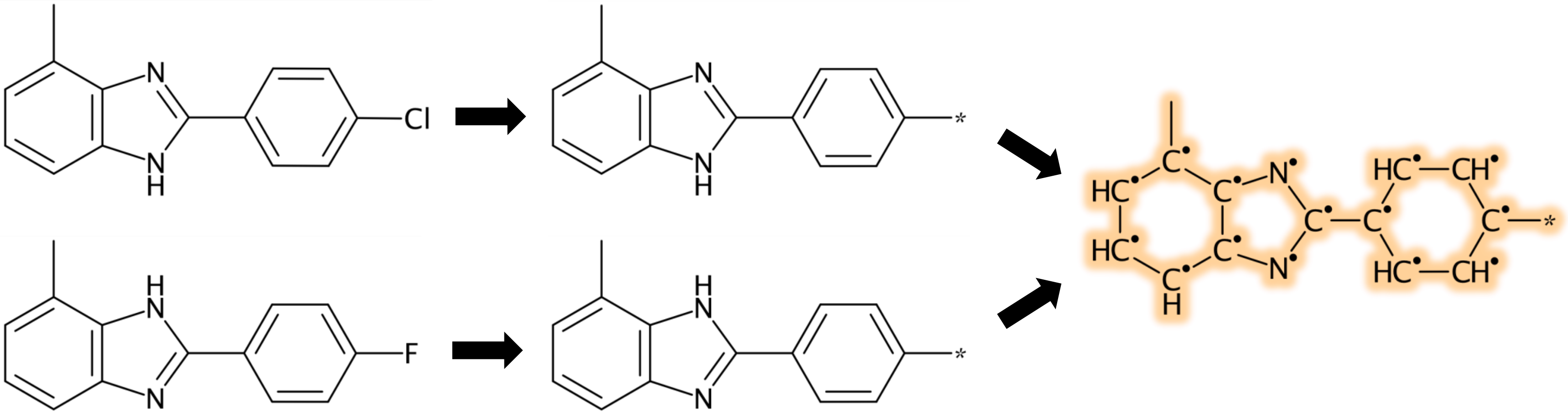


C[C]1[CH][N][CH][N]1_1

- Two parts: a SMILES component and an integer
  - The canonical SMILES after setting all bonds to single bonds, all charges to zero, and stripping hydrogens from heteroatoms
  - An integer: the count of stripped hydrogens minus total charge
- Identify duplicate structures independent of charge, tautomeric state or resonance form

APPLICATIONS

IDENTIFY MATCHED PAIRS II



- Replace each R group in turn by an asterisk, and generate the ~~canonical SMILES~~ tautomer hash
- Matched pairs share the same hash, independent of tautomeric form

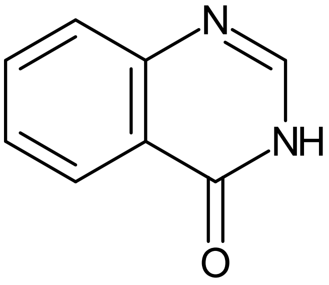
TAUTOMER NORMALIZATION BASED ON MEDICINAL CHEMISTS

- Need a source of structures exactly as drawn by chemists
  - US Patents - ChemDraw files
  - ChEMBL – those structures taken from journals (*)

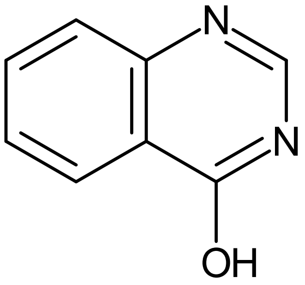
* ChEMBL retains the first tautomeric form entered for any molecule. For structures taken from journals, the tautomer as present in the journal reference is used. However, ChEMBL also incorporates other data sources and structures from those sources may have been normalised.

TAUTOMER NORMALIZATION BASED ON MEDICINAL CHEMISTS

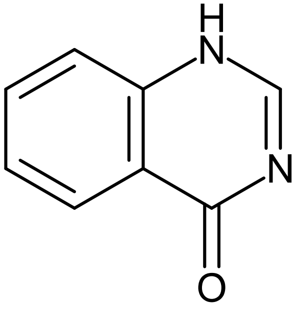
- Need a source of structures exactly as drawn by chemists
- Identify all tautomeric regions* and generate the corresponding heteroatom protomer hash



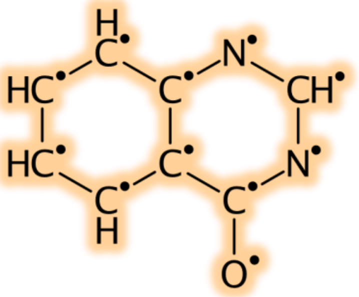
420



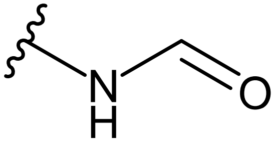
959



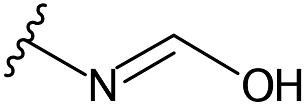
3



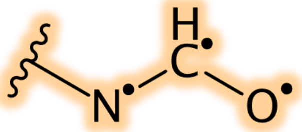
[O][C]1[N][CH][N][C]2[CH][CH][CH][CH][C]12_1



540368



52

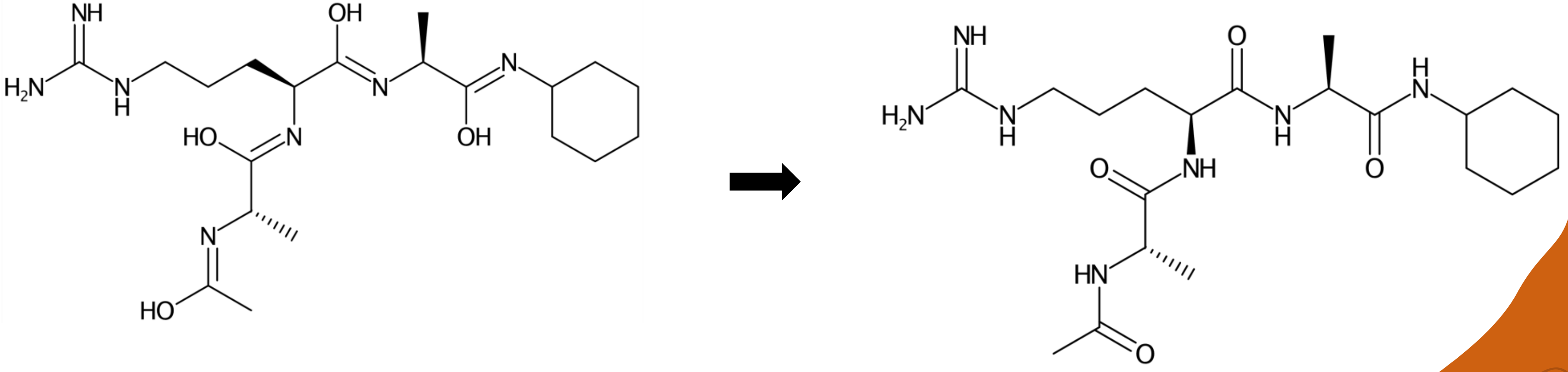


*[N][CH][O]_1

* R. Sayle, J. Delany. *Canonicalization and Enumeration of Tautomers.* **EuroMug99**, Cambridge, UK. https://www.daylight.com/meetings/emug99/Delany/taut_html/index.htm

TAUTOMER NORMALIZATION BASED ON MEDICINAL CHEMISTS

- Need a source of structures exactly as drawn by chemists
- Identify all tautomeric regions and generate the corresponding heteroatom protomer hash
- First the most popular form for each tautomeric substructure, and use to normalize databases (or flag up dubious structures)



CHEMBL464806

DATABASE SEARCHING

- Add hashes as columns to enable:
  - Tautomer independent searches
  - Searching for molecules with the same Murcko scaffold
  - Searching for matched pairs of a molecule
  - (…etc.)

- Exploit any subset relationship between hashes
  - E.g. all element graphs have the same anonymous graph (but not *v.v.*)
  - Similarly, extended Murcko/Murcko, regioisomer/MF, tautomer hash/protomer hash
  - Only store the more general form in a database column, and calculate the more specific one on-the-fly

OTHER LINE NOTATIONS ARE AVAILABLE

- InChI
  - A molecular hash independent of atom order and tautomeric form
- CACTVS Hash codes, and NCI/CADD structure identifiers
- Reduced graphs (Sheffield, plus others)
  - Text representation where similar ring systems have the same representation, is invariant to ring system locant (e.g. Birchall et al. *JCIM*, **2009**, *49*, 1330 and refs therein)
- Related work:
  - S. Urbaczek, A. Kolodzik, M. Rarey. The valence state combination model: a generic framework for handling tautomers and protonation states. *JCIM*, **2014**, *54*, 756.

CONCLUSIONS

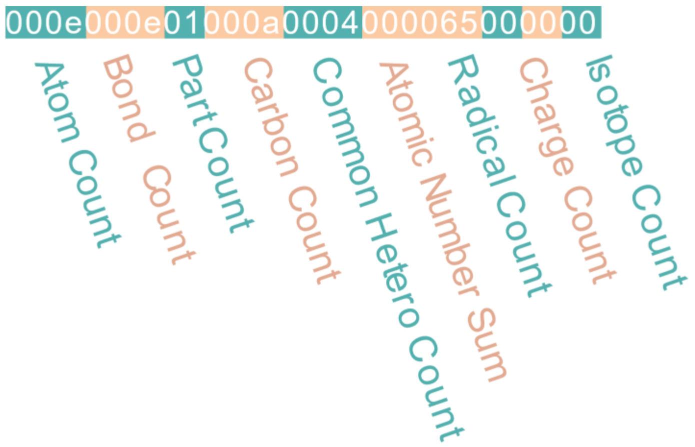
Take-home message

- If the solution appears to require enumeration, think about whether a hashing approach is possible
- Molecular hashes can be used to efficiently investigate many problems in cheminformatics
- You just need the right hash!
  - What information to leave out?

Acknowledgements

- John Mayfield
- Jonas Boström
- John Overington
- Oliver Korb

ARTHOR SUBSTRUCTURE INDEX HASH



000e000e01000a0004000065000000

Atom Count

Bond Count

PartCount

Carbon Count

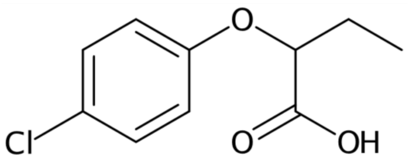
Common Hetero Count

Atomic Number Sum

Radical Count

Charge Count

Isotope Count



- A hash that can be used to sort a database to be searched
- Ensures that substructure search results will be returned in an order that approximates similarity to the query
  - Smallest molecules first
  - ‘Plain’ molecules first – no radicals, charges or isotopes