Chapter 11
Search Engine for Chemical Molecules

Introduction / motivation

In this chapter we are going to discuss possible rankings of molecules in order to facilitate search through molecule names and formulas. Our main goal is to give a chemist the molecules he is looking for with minimal number of symbols typed in.

The main obstacle to achieve this goal is that there are too many molecules that would fit a query. For example, for the search string "Me" *Mathematica* has nearly thousand molecules that start with that string:

```
Select[ChemicalData["*"], StringMatchQ[#, "Me" ~~ ___] &] // Length
```

```
967
```

If we search through the chemical formulas there are nearly three thousand molecules, the formulas of which begin with "C2":

```
With[{cnames = ChemicalData["*"]},
  Select[ChemicalData[#, "CompoundFormulaString"] & /@ cnames,
   StringMatchQ[#, "C2" ~~ ___] &]] // Length
```

```
2963
```

One way to tackle this is problem is to assign a (static) rank to each molecule, which quantifies its significance or importance related to the rest of the molecules, and order the search results according to these molecule ranks. This approach is analogous to the one taken with the *PageRank* algorithm used by *Google*. The *PageRank* algorithm calculates the ranks of web pages using the links between them. So our goal is first to find ways of linking molecules. Then we are going to define prestige and rank of a molecule, and show that they lead to eigenvector computation of a matrix.

Molecule linking

We want to find methods of linking molecules to each other that would imitate the way a curious chemistry student would browse through them, or an experienced chemist would have a systematic knowledge of them.

Another way to look at the molecule linking is that with it we are trying to simulate a chemist's random molecule browsing.

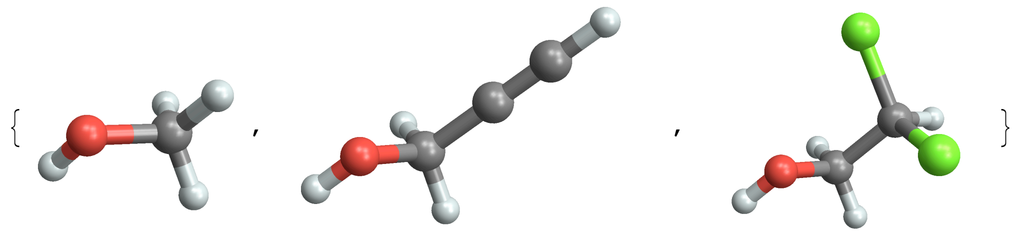
There are several (probably many) possible ways to connect molecules:

1. Through graph intersections. If the graph of a molecule is a sub-graph of another one then we link the first one to the second. We can also make that link if a large (or significant) part of the graph of the first is embedded in the graph of the second.
2. Since the search is done through symbols that are typed in, it would make sense to link molecules according to the symbols in their compound formulas or in their representation with any serial/line notation for molecules (like SMILES).
3. We can link molecules if they have common or analogous physical and chemical properties.
4. We can use all of the above methods thorough some combination procedure.

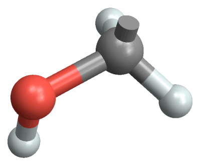
Probably the most interesting and appealing method is the first one, which uses molecule graphs. Graph intersection and embedding checking, though, can be hard to develop and computationally expensive. (Probably NP-complete.) So we can take a similar, but simpler approach, with which we compare the number of types of bonds in the molecules.

Linking through bond tallies

Consider the 3D plots of the molecules Methanol (CH_4O), PropargylAlcohol (C_3H_4O), 2,2Dichloroethanol ($C_2H_4Cl_2O$):



The molecules have the common part:



and they also share an additional carbon-hydrogen connection.

The similarities of the molecules can be also seen if we look at their bond tallies:

```
{{{H, O}, 1}, 1}   {{{C, O}, 1}, 1}   {{{C, H}, 1}, 3}

{{{H, O}, 1}, 1}   {{{C, O}, 1}, 1}   {{{C, H}, 1}, 3}   {{{C, C}, 3}, 1}    {{{C, C}, 1}, 1}

{{{H, O}, 1}, 1}   {{{C, O}, 1}, 1}   {{{C, H}, 1}, 3}   {{{C, Cl}, 1}, 2}   {{{C, C}, 1}, 1}
```

We can define a similarity function that finds percentage of common bond tallies compared to the bond tallies of the first molecule:

```
BondSimilarity[cname1_String, cname2_String] :=
  Block[{t1 = ChemicalData[cname1, "BondTally"], t2 = ChemicalData[cname2, "BondTally"]},
   Total[Intersection[t1, t2]⟦All, 2⟧] / Total[t1⟦All, 2⟧]
  ];
```

and use that function in functions like `Nearest` or `FindClusters`:

```
Nearest[cnames, "Methanol", 5, DistanceFunction → (1 - BondSimilarity[#1, #2] &)]

{Methanol, PolyvinylAlcohol, PropargylAlcohol,
 2,2Dichloroethanol, 2,2,3,3Tetrafluoro1Propanol}
```

Prestige and rank of a molecule

To be written...

Computing the eigenvector

To be written

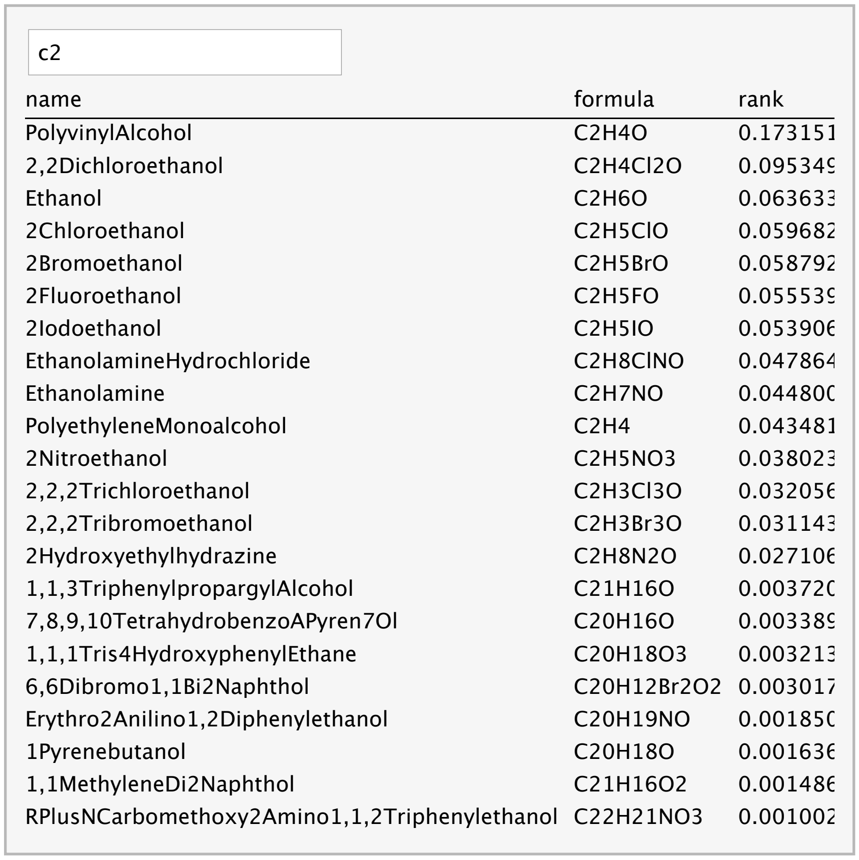
Search example

Below is implemented molecule name and formula searching with rank ordering for the alcohol molecules. In the interface below one can use both molecule names and chemical formulas. (Try "me" and "C3".) The search string is matched with the beginning of the names or formulas.

To try out the implementation evaluate the sub-sections below.

Alcohols ranking data from bond tallies (evaluate this)

Interface (evaluate this)



c2

name	formula	rank
PolyvinylAlcohol	C2H4O	0.173151
2,2Dichloroethanol	C2H4Cl2O	0.095349
Ethanol	C2H6O	0.063633
2Chloroethanol	C2H5ClO	0.059682
2Bromoethanol	C2H5BrO	0.058792
2Fluoroethanol	C2H5FO	0.055539
2Iodoethanol	C2H5IO	0.053906
EthanolamineHydrochloride	C2H8ClNO	0.047864
Ethanolamine	C2H7NO	0.044800
PolyethyleneMonoalcohol	C2H4	0.043481
2Nitroethanol	C2H5NO3	0.038023
2,2,2Trichloroethanol	C2H3Cl3O	0.032056
2,2,2Tribromoethanol	C2H3Br3O	0.031143
2Hydroxyethylhydrazine	C2H8N2O	0.027106
1,1,3TriphenylpropargylAlcohol	C21H16O	0.003720
7,8,9,10TetrahydrobenzoAPyren7Ol	C20H16O	0.003389
1,1,1Tris4HydroxyphenylEthane	C20H18O3	0.003213
6,6Dibromo1,1Bi2Naphthol	C20H12Br2O2	0.003017
Erythro2Anilino1,2Diphenylethanol	C20H19NO	0.001850
1Pyrenebutanol	C20H18O	0.001636
1,1MethyleneDi2Naphthol	C21H16O2	0.001486
RPlusNCarbomethoxy2Amino1,1,2Triphenylethanol	C22H21NO3	0.001002

"More like this molecule"

Using the similarity function with which the links are made we can provide a "more like this" functionality for a molecule in the search result. (See the example with `Nearest` above.) We can give nearest neighbors of for a search result molecule or a cluster it belongs to. The similarity function used can be also based on the molecules physical and chemical properties.

Extensions / Exercises

Possible extensions that are relatively easy to implement are:

1. Computing of dynamic ranks instead of static ranks. (Analogous to *ExpertRank* used by *Ask.com*.)

2. Natural language queries through relevance finding using vector space model for molecule property pages.