An early structure diagram
Kekule’s representation of benzene

Molecular Informatics

http://www-jmg.ch.cam.ac.uk/cil/partii/

Professor Robert Glen (rcg28@cam.ac.uk)
Summary

Chemists today are expected to deal with vast numbers of molecules. Not just those made, but those that could be made. Some, as yet undiscovered, may cure cancer, catalyse new reactions or have amazing structural qualities.

In your career, you don’t just want to make molecules, but to make the few that really work. In companies, billions of pounds are being invested in information technology to do just this.

What methods are being developed and how are they used? How will Artificial Intelligence impact your future work?

We hope to give you some insight into how molecules are stored on the computer, what properties of molecules can be computed and how data on millions of molecules is stored and searched.

There are two lectures

1 - Molecules and Computers - how do we store molecules on the computer and how do we access the data?

2 - Finding Molecules and calculating their properties - how do we search and retrieve one molecule from a million?

This is followed by a series of exercises to aid understanding of Molecular Informatics/Cheminformatics.
Introduction to Molecular Informatics/Cheminformatics and Modelling

Molecular Informatics: includes all aspects of the study of molecules on computers. This includes the representation of molecules, databases, display, simulation, prediction of their properties and the discovery and design of new molecules and materials.

Note: the online version of this document contains many hyperlinks to other related content.

History

Since the dawn of the age of computers, chemists have simulated molecules! Pioneers coupled engineering display devices to primitive computers to display and manipulate molecules. We built our first modelling system from a Tektronix Display coupled to a PDP-11.

We used molecular mechanics with x-ray structures of small molecules and ‘overlaid’ them to deduce common features and related these to observed properties.

Nowadays we use supercomputers and 3-D graphics

There are a number of articles that describe the development of Molecular- (or Chem-) Informatics, some examples are:

Cheminformatics: a history

Cheminformatics: Past Present and Future

Computational chemistry and cheminformatics: an essay on the future

Cheminformatics in Drug Discovery – includes AI approaches
There have been a number of books published which will introduce you to the subject in greater depth:

**An introduction to chemoinformatics** (Leach & Gillet) – a good starting point

**Chemoinformatics – basic Concepts and Methods**

**Chemoinformatics: An Approach to Virtual Screening By Alexandre Varnek, Alex Tropsha, RSC Publishing**


Molecular informatics is closely related to bioinformatics, computational chemistry, molecular modelling, simulation, machine learning and statistics - but the area has principally been driven by investment in new methods for drug discovery, hence the concentration on small organic molecules. There are a large number of journals devoted to or make extensive use of Molecular informatics methods:

- **Journal of Chemical Information and Modeling** – historically, this has been the major journal
- **Journal of Chemical Theory and Computation**
- **Journal of Cheminformatics**
- **Journal of Computer-Aided Molecular Design**
- **Journal of Molecular Graphics & Modeling**
- **Journal of Computational Chemistry**
- **Journal of Medicinal Chemistry**
- **Reviews in Computational Chemistry**
- **Drug Discovery Today**
- **BMC Bioinformatics**
- **Nature Reviews Drug Discovery**
- **Expert Opinion on Drug Discovery**
- **WIREs computational Molecular Science**

The “informatics” (or “data analytics” or “data Science”) area is especially important with the advent of huge amounts of chemical information (“Big Data”) – informatics is a broad academic field encompassing computer science, human-computer interaction, information science, information technology, algorithms, areas of mathematics (especially mathematical logic and category theory), and social sciences – and we take advantage of all these areas and apply the methods to our ever growing molecular databases.

**Chemistry and mobile apps.** A large number of mobile apps for pads and phones are appearing. Many of these are itemised [here](#) or [here](#). As with all software, explore the app with some known examples to check its reliability and accuracy (e.g. it’s not unknown for molecular weights to be wrongly calculated or for stereochemistry to be inverted or randomised).

**Storing and finding molecules on a computer**

There are guestimated to be $10^{60}$ possible small molecules that you could make. How do you find the best molecule for the problem you are addressing? Let’s take a look “under the bonnet” of the way molecules are actually manipulated on the computer. This will help you understand the limitations, sources of error and abilities of current methods.
You are probably most familiar with Chemical nomenclature (IUPAC) which requires a deep knowledge of chemistry and standards e.g. Morphine (which is a trivial name) is (using IUPAC nomenclature) \( (5\alpha,6\alpha)-7,8\)-didehydro-4,5-epoxy-17-methylmorphinan-3,6-diol while cocaine is methyl\(1R,2R,3S,5S\)-3-(benzoyloxy)-8-methyl-8-azabicyclo[3.2.1]octane-2-carboxylate (you could try our (we think) very clever program OPSIN which converts names to structures). However, representing chemicals in this way does not easily facilitate computer storage and retrieval. For example, finding two molecules that have the same chemical fragments (both morphine and cocaine act at the same biological receptor, have a phenyl group and a protonated amine at physiological pH). To address this, molecules are stored using different approaches for different purposes. They can conveniently be broken down into 1-Dimensional, 2D and 3D descriptions.

1-Dimensional : line notations
A 1D - representation of a molecule is a string that contains the valence bond description of a molecule. Past examples include Wiswesser Line Notation and Rosdal. Examples for this molecule:

- Line Notations
  - WLN

The most widely used line notation is SMILES. The big advantage of SMILES is that it is easy to learn, can be ‘parsed’ very quickly by a computer and is very compact – and is human interpretable. The entire language can be learned in a few hours. The disadvantages are that it is only for organic small molecules, it does not include tautomers, atropisomers, non-covalent bonds and a few other omissions.

Here are examples of SMILES:
Methane C, Ethane CC, Methanol CO, L-alanine N[\(\text{C@H}\)][\(\text{C}\)](=O)O. You may have noticed that ethanol can be CO or OC, so “canonicalization” is necessary, which are rules to ensure SMILES are always the same for one molecule, no matter how the SMILES is produced.

As you can see, SMILES uses the keyboard symbols to describe molecules. A tutorial on SMILES is given here: [http://www.daylight.com/dayhtml_tutorials/languages/smiles/index.html](http://www.daylight.com/dayhtml_tutorials/languages/smiles/index.html). SMILES is widely used in computer databases to store structures. You can practice SMILES using the Mollinspiration webpage. There is an extension of SMILES called SMARTS which allows the storage of reactions and more complex queries of the database. e.g. C(=O)O.OCC>>C(=O)OCC.O would match an esterification reaction.

Another more modern line notation is called InChI. This addresses some of the problems of SMILES and is being expanded to polymers and materials. InChI is generated using computer algorithms and is virtually un-interpretable by a human (although with skill it can be deduced). Morphine is InChI=1S/C17H19NO3/c1-18-7-6-17-10-3-5-13(20)16(17)21-15-12(19)4-2-9(14(15)17)8-11(10)18/h2-5,10-11,13,16,19-20H,6-8H2,1H3/t10-,11+,13-,16-,17-/m0/s1. It is commonly used as a unique
chemical identifier - each molecule should theoretically have a unique InChI – although some clashes have been found. Websites (e.g. Chemspider from RSC) are available that can generate InChI from different names and formats. Again, a string like this is easily matched on a computer.

2- and 3-Dimensional molecules: connection tables and coordinates
Sometimes we wish to store and search chemical structures (basically their chemical diagram). You may wish to draw a structure and search for a starting material for a reaction. There are many databases of this type (ignoring the three dimensional structures of molecules). Examples are PubChem, Chemspider, eMolecules.
The valence model of a molecule can be represented by a graph. A simple graph contains nodes (atoms) and edges joining pairs of nodes (bonds).

The spatial position of the nodes, length of the edges and crossings are irrelevant. Generally, we ignore hydrogens unless tautomerism or pKa is an issue. Computers handle graphs very well and molecules represented like this are examples of labelled graphs (atoms are labelled oxygen, carbon etc.).
Chemical structures are of course more complex than this and aromaticity, stereochemistry, tautomersim, non-stoichiometric compounds etc. are often problematic. The computer would deduce these two canonical structures are different molecules!
So we introduce ‘fixes’ to ensure these would map on to each other e.g. create the concept of an aromatic bond.
Here is an example of a labelled graph format commonly used to describe molecules (the SD file) – for benzene. In true ancient FORTRAN tradition, each line and fixed range of the line has an expected meaning.

benzene
ACD/Labs0812062058
August 2013
6 6 0 0 0 0 0 0 0 0 1 V2000
1.9050 -0.7932 0.0000 C 0 0 0 0 0 0 0 0 0 0 0 0
1.9050 -2.1232 0.0000 C 0 0 0 0 0 0 0 0 0 0 0 0
0.7531 -0.1282 0.0000 C 0 0 0 0 0 0 0 0 0 0 0 0
0.7531 -2.7882 0.0000 C 0 0 0 0 0 0 0 0 0 0 0 0
-0.3987 -0.7932 0.0000 C 0 0 0 0 0 0 0 0 0 0 0 0
-0.3987 -2.1232 0.0000 C 0 0 0 0 0 0 0 0 0 0 0 0
2 1 1 0 0 0 0
3 1 2 0 0 0 0
4 2 2 0 0 0 0
5 3 1 0 0 0 0
6 4 1 0 0 0 0
6 5 2 0 0 0 0
M END
$$$$
Here is a description of what each line contains.

<table>
<thead>
<tr>
<th>Line</th>
<th>Section</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Molecule name</td>
<td>&quot;benzene&quot;</td>
</tr>
<tr>
<td>2</td>
<td>User/Program/Date/etc information</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Comment (date here)</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Counts line</td>
<td>6 atoms, 6 bonds, ..., V2000 standard</td>
</tr>
<tr>
<td>5-10</td>
<td>Atom block</td>
<td>(1 line for each atom): x, y, z, element, etc.</td>
</tr>
<tr>
<td>11-16</td>
<td>Bond block</td>
<td>(1 line for each bond): 1st atom, 2nd atom, type of bond, etc.</td>
</tr>
<tr>
<td>17</td>
<td>Properties block</td>
<td>(empty)</td>
</tr>
<tr>
<td>18</td>
<td>$$$$</td>
<td>Terminator line (we may have a list of many molecules)</td>
</tr>
</tbody>
</table>

So, we have the number of atoms (6), the number of bonds (6), their coordinates (which are not necessary for 2D use, e.g. structure search) and the atom type. The unfilled portions of the atom description (zeros) can indicate stereochemistry, isotope, ionisation etc. The “bond matrix” contains “from atom” – “to atom” connectivity and the bond type, 1 for single and 2 for double etc. Some more properties of bond matrices are important. Here is a simple molecule and its bond matrix (as above) e.g. atom 5 bonded to atom 6 with a double bond can be seen in the matrix.

**Bond matrix**: indicates which atoms are bonded, and the corresponding bond orders.

```
    1  2  3  4  5  6
1   0  1  0  0  0  0
2   1  0  1  0  0  0
3   0  1  0  1  1  0
4   0  0  1  0  0  0
5   0  0  1  0  0  2
6   0  0  0  0  2  0
```

Alternatively, we can use an adjacency matrix (atom 3 is bound to 2, 5 and 4)

**Adjacency matrix**: indicates which atoms are bonded.

```
    1  2  3  4  5  6
1   1  1  0  0  0  0
2   1  0  1  1  1  0
3   0  1  0  1  1  0
4   1  0  0  0  0  2
5   0  0  0  0  0  2
6   0  0  0  0  2  0
```

Bond matrices are common for small molecule representations, however for proteins as an example (which are large molecules where typically only the atom positions are determined, not the bonds),
an adjacency matrix is used (in what is called a PDB file – Protein Data Bank File) using the CONNECT record (lines in the data). So we could have “Connect 3 2 4 5” meaning atom 3 is connected to atoms 2, 4, 5. An example of a xxx.PDB file is shown below.

```
HEADER  OXIDOREDUCTASE/OXIDOREDUCTASE INHIBITOR  26-MAY-11  2SA
TITLE  HUMAN DIHYDROFOLATE REDUCTASE BINARY COMPLEX WITH PTO4
EXPTLA  X-RAY DIFFRACTION
REMARK 2 RESOLUTION.  1.80 ANGSTROMS.
REMARK 200  TEMPERATURE (KELVIN) : 113
REMARK 200  PH : 6.9
SEQRES  1  A  116  VAL GLY SER LEU ASN CYS ILE VAL VAL VAL GLN ASN
SEQRES  2  A  116  MET GLY ILE GLY LYS ASN GLY ASP LEU PRO TRP PRO PRO
CRYST1  54.588  55.106  64.827  90.00  90.00  90.00  P 1 2 1 2 1 4
ORIG1   1.000000  0.000000  0.000000  0.000000
ORIG2   0.000000  1.000000  0.000000  0.000000
ORIG3   0.000000  0.000000  1.000000  0.000000
SCALE1  0.000000  0.000000  0.000000  0.000000
SCALE2  0.000000  0.000000  0.000000  0.000000
SCALE3  0.000000  0.000000  0.000000  0.000000
ATOM    1  N  VAL A 1   3.836  -1.035  -3.583  1.00  31.28
ATOM    2  CA VAL A 1   4.283  -0.343  -3.015  1.00  29.14
ATOM    3  C  VAL A 1   4.565   1.057  -3.583  1.00  27.64
ATOM    4  O  VAL A 1   4.663   1.287  -4.863  1.00  27.17
ATOM    5  CB VAL A 1   6.310  -1.245  -3.141  1.00  30.01
.

eutron  1910  S  SO4 A 187  28.393  -1.362   5.883  1.00  22.94

eutron  1911  O1  SO4 A 187  28.504  -1.159   4.508  1.00  22.13

eutron  1912  O2  SO4 A 187  27.282  -0.770   6.155  1.00  24.10

eutron  1913  O3  SO4 A 187  24.953  -1.035   6.850  1.00  15.34
.
CONECT  1510  1511  1512  1513  1514
CONECT  1511  1510
CONECT  1512  1510
.
MASTER   334   0   6   6  18   0  11   6  1533   1  95 16
END
```

(note: in x-ray structures of proteins, we can only “see” the heavy (non-hydrogen) atoms, and not the bonds, so we infer the bonds from the atom positions. Sometimes the resolution is too low, which causes problems in identifying atoms never mind bonds e.g. the oxygen and nitrogen in the asparagine sidechain is sometimes confused and incorrectly labelled). Here is an electron density map determined from x-ray crystallography – the atom positions and the bonds are modelled into the density.

Which way round should this ring go?
Problems that have been addressed are aromaticity (having a special aromatic bondtype), stereochemistry (having a special flag that determines whether a bond is up or down from the plane of the attached atoms), tautomers (by storing copies of the same molecule, but with different tautomers) and also by canonicalization (rules):

- e.g. a nitro group can be stored as the ionic or hypervalent form – we decide on the rule and stick to it (but you need to know the rules when you do a search!).

What if we want to vary the atom positions e.g. in a reaction?

Another file format which is still in general use is the Z-Matrix. In this format, we describe a molecule by its bonds, angles, torsions and non-bonded distances (known as internal coordinates). This is particularly useful to create e.g. reaction coordinates, where an atom is moving or a torsion angle is changing. Using Cartesian coordinated would be much more difficult. Here are the elements of a typical Z-Matrix:
And here is a typical Z–Matrix (for methanol).

![Z-matrix diagram]

We start with Carbon atom 1 (it’s on line 1), next line it says this Oxygen atom 2 (it’s on line 2 etc.) is joined to atom 1 with a bond length of L1 (1.42, next block of characters), atom three Hydrogen is joined to atom 1 and has a bond length of L2 and makes a bond angle for atoms 3-1-2 of a1 (109.0). The next atom H4 (line 4) of course also needs a torsion (dihedral) angle (work that one out – it’s a bit special, as it is an improper torsion angle). Internal coordinates are good for small molecules, but for large molecules, errors tend to accumulate as everything is referenced from the first atom.

Chemical reaction calculations from a reactant to a product via a transition state can be driven using internal coordinates which alter bond lengths, angles and non-bonded distances for example.

3-Dimensional molecules

Molecules are of course not typically flat. Thermal fluctuations mean atoms in real molecules move. So we represent 3D molecules by including their coordinates or their internal coordinates. Notice in the benzene example above, that all the Z-coordinates are zero, but in reality there is at least thermal fluctuation in the atom positions. Always remember this when you look at a molecular picture – it is a static representation of one geometry and conformation. Getting the 3-dimensional coordinates can involve experiment (x-ray, electron or neutron diffraction e.g. the [Cambridge Crystallographic Database](https://www.ccdc.cam.ac.uk) or the Protein Databank – PDB). From these can be obtained atom positions, bonds, coordinates etc. There are a number of 3-D construction methods available such as [MolSoft](https://www.molsoft.com) (put in a SMILES and get a 3-D molecule) Corina and Concord. Molecules can also be constructed in 2D and subjected to molecular mechanics or Quantum Mechanics calculations to obtain 3D structures. Conformation still remains to be deduced, and will depend on the environment of the molecule (e.g. changing solvent or non-covalent binding to a substrate can alter...
There are many methods that deduce conformations, usually involving torsional angle rotation to scan the conformational space (like a Ramachandran plot – but often in many more torsional angle dimensions).

The latest way of storing molecules in databases uses a new file type called XML (eXtensible Markup Language). A language called Chemical Markup Language (CML) has been developed to store and access molecules, their coordinates and their properties (or metadata as it is called). XML is slowly replacing other forms of data storage due to its flexibility. One advantage is the extensible nature of the format (we can easily add metadata) – metadata is data that describes data – e.g units such as degrees Celsius, how this is related to degrees Kelvin, and what a probable range of reasonable values could be, who collected the data (and on what day – maybe Friday afternoons after 4.00 o’clock includes more error prone data) and also ontologies – descriptions of the relationships between data (e.g. we use subject/predicate/object terminology to describe entities (i.e. anything) e.g. a molecule (subject) contains an element (predicate) carbon (object). These ‘triples’ can be searched. E.g. find all the molecules that contain carbon. It moves the question and answer into the linguistic realm using English (or French or whatever – they can be translated).

“With RDF, knowledge is represented in terms of subject-predicate-object triples, where each member of a given triple may be a dereferenceable Universal Resource Identifier (URI) for a particular concept or entity. Thus, what was traditionally referred to as a database, could now be considered a knowledge base, as the initially inert data points were given a machine-understandable meaning through reference to formally specified concepts in supporting ontologies. Thus, two entries from two different knowledge bases could be inferred to relate to the same concept even if no such inference had been explicitly stated, through machine reasoning over axioms in the supporting ontologies.”

Some interesting concepts in this article will give an insight into modern ways of storing chemical data (if you are interested).

File Format interchange
There are many file formats for molecules. There is however a freely available program called OpenBabel which will convert most of these to each other. You can download a Windows version here.

Dynamics of Molecules
Molecules move, and we can model this using a methods such as molecular dynamics. This method simulates the motions of atoms as particles in a Newtonian mechanics force-field, it is relatively fast and can now (using supercomputers) simulate large systems. Informatics is used to analyse the trajectories – how the molecules change in a time-dependent fashion. The methodology is used to understand material properties, in drug discovery, in biological systems, or in self-assembly of molecular clusters as examples. The very large datasets produced result in the need to analyse the “Big Data” data. Here is an interesting example using graphene to sequence DNA, with an associated simulation of the DNA progressing through the nanopore. Of course with such large datasets, we can do statistical analysis (e.g. correlated motions of parts of the system), analyse the key interactions (contact maps), look at the formation of binding sited (e.g. in proteins for drug discovery) etc. or compute thermodynamic properties (e.g. melting temperature of polymers).
Finding Molecules
Supposing you have access to a database of 100-million molecules (like Sci-Finder)

- CAS REGISTRY contains more than 106 million unique organic and inorganic chemical substances, such as alloys, coordination compounds, minerals, mixtures, polymers and salts, and more than 65 million sequences and we perform a search. In about 0.1 of a second, answers appear – why is this so fast and how is it done? Clearly not all the molecules are being searched. The secret to this lies in “keys”. That is, pointers to all the molecules that you don’t need to search.

Hashing
The molecular graph of a molecule (the atoms and bonds) contains most of the information to find a molecule. From this we can deduce some simple search rules. E.g. suppose we create a set of “keys” that represent the presence or absence of atoms, fragments etc. We could say if phosphorous or a ketone were present or not in a molecule. If we then have a list of ones and zeros (a binary string) we can say whether or not a fragment is present or not in a molecule.

[Diagram of molecular graph with keys and binary string]

If we pre-process the entire database, when we do a search for a molecule that e.g. contains phosphorous, we can eliminate all the molecules that don’t contain phosphorous in a flash. This is called hashing. Computers are very fast at matching bit strings. We can even intelligently partition the database into subsections that can be ignored before we even start – e.g. put all the unusual molecules (sugars? boron-containing?) into different datasets.

We can also generate fragments automatically using “fingerprints” which are special fragments that we will come on to later.

Next step – Pattern Matching
Now we have the molecules down to a smaller set, we can do the difficult part which is to match the pattern of the query molecule to what remains of the database.

Exact match. We can now match two canonicalised SMILES strings or a pair of InChI keys (see above or hyperlink) to find a molecule that is an exact match to the query, effectively matching ID’s. Of course, remember that for molecules that have e.g. different salts, you may be interested in only the larger portion of the molecule – so some care needs to be taken to find all the possible answers. Sometimes we can just strip out all the salt counterions (usually called ‘washing’), but on occasion this can cause problems as we effectively have different compounds that are now being classified as the same in the answer list.
Substructure Search. Supposing we wish to find part of a molecule in a database. To do this we have to do a substructure search. A substructure is a sub-graph of the molecular graph. This is an example of substructural fragments of a larger molecule. You could e.g. look for all the molecules with the phenol fragment. There are two steps commonly used to do this, firstly we have to number all the molecules consistently, for example to compare two structures to see if they are identical (the Morgan algorithm) then we have to match the query substructure to each molecule (the Ullman algorithm).

Morgan Algorithm. The Morgan algorithm iteratively computes an integer label for each node in a structure (atoms). It works by:

(i) Assign an integer label $i$ to each node considering its atomic number, degree (number of substituents) and types of bonds

(ii) Update for each label $i$ by adding up the labels of the adjacent nodes (connected atoms)

(iii) Repeat (ii) until the number of different labels does not increase. Then order the nodes by the value of the labels


This example below takes the molecule and reduces it to a graph, then each point has the number of connections assigned, then we iterate over the graph adding the number from each of the connected atoms till we get no change (notice symmetry related atoms cannot be ‘disambiguated’). Then we number from the largest to smallest to get a numbering scheme and rules are used to break ambiguities such as N before C, or double bonds or more substituted atoms have higher priority.

After numbering we need to match the substructures. Each atom can only have one number, hence the need to number the system (otherwise we would have a peculiar bond matrix).

Ullman algorithm. This is a way of detecting subgraph isomorphisms (fragments) – parts of molecules that are the same e.g. two benzene rings.

We use matching of adjacency matrices to find the subgraph (fragment) (the two shaded regions).
This is slow, as it involves matching all possible matches to find a fit (termed an np-complete problem) which why we hash and filter to speed up the process and only match the subgraphs at the final stage. This matching process can be very difficult with a common type of molecular representation called Markush structures.

**Markush structures**

A curious name and it comes from the first chemical patents of generic structures. In 1924 Dr. Markush was awarded a patent on pyrazolone dyes (USP No. 1,506,316) in which he claimed generic chemical structures in addition to those actually synthesized. Structures of this type were permitted after a ruling in 1925 by the US Patent Office and became known as “Markush structures”. The “Markush Doctrine” of patent law greatly increases flexibility in the preparation of claims for the definition of an invention. Notice the “R” groups, these can be defined as anything, and multiplication of these together can result in millions of structures. Here is an example:
Searching Markush structures is still an ongoing problem. (Markush structure searching over the years, Edlyn S. Simmons World Patent Information, Volume 25, Issue 3, September 2003, Pages 195-202 ... and not much has changed since then). Marpat and Questel are commercial databases with Markush searching. If you want to patent a new molecule, it needs to be completely novel, and therefore it needs to be checked to see if it has been claimed before - even in Markush structures, of which there are billions. It is not unknown to miss a structure, which then invalidates the patent.

Finding molecules using Molecular Similarity. Sometimes the question of finding new molecules can be more ambiguous e.g. “find me a molecule that is like morphine”. There is much debate as to what precisely “like” means - it obviously has a context. If we ask for a molecule like limonene, do we mean one that is structurally similar, is made in a similar way, or smells like limonene? It is not unknown to miss a structure, which then invalidates the patent.
Here are some ways of finding similar molecules.

Maximal Common Subgraph (MCS). This is an important search to determine the largest part of a structure that is constant between the query and molecules in the database. The MCS of a reaction is shown here (the atoms and bonds which are constant). We can use this to find e.g. reagents with a common reactive group. This is a complex case of identifying a fragment, as we don’t know the size of the MCS beforehand, so involves
‘backtracking’ to compute – and is therefore time consuming. The algorithm iterates over ever larger
regions of the molecule until a match is lost, and does this for each atom. The largest match is the
MCS. For example, this is also one of the problems of converting a list of compounds to a Markush
structure. Structural similarity between compounds can be based on e.g. the number of common
atoms in the MCS – so we could look for molecules that contain at least 80% of the query structure.

**Similarity using molecular fingerprints.** Molecules can be “fingerprinted” from their connection
tables. This results in a “bar-code” for the molecule. This is a string of ones and zeroes, and we know
computers are very fast at matching these. Comparing the bar-codes gives a measure of similarity.
The simplest way to fingerprint a molecule is to generate a **hash code** (see earlier) using the
presence or absence of fragments. This is used in the CCDC x-ray database software.

- Hash codes (already mentioned for searching)
- The simplest fingerprints

```
• Doesn’t contain F
• Contains P

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
```

Fingerprints can be generated automatically (so we don’t have to manually define all the interesting
substructures).
Most methods rely on variations of the **CRC** algorithm. First, the molecule is broken down into
(typically) 4-7 atom linear fragments, these are then assigned a number (could be from adding their
atomic weights), then this is divided into a large prime number, which gives a remainder (the
remainder from the largest number that divides into the prime defines the length of the string). This
location in the string is set to one. This is repeated for all the defined fragments in the molecule,
resulting in a ‘bar-code’. 
Another way to do this is to use circular fingerprints. These are based on the connectivity shells around single atoms and describe the environment around atoms. The 'atom types' (e.g. carbon sp$^3$ shown as C.3 here) describe the valency and hybridisation of the atom. The bit string is made up of concatenated layers of lists of the atom types which are pre-defined. Usually we go down three layers.

Comparing fingerprints to deduce similarity. There is a wide literature in computer science in comparing bit strings, and there are many flavours of bit-string comparisons used in Cheminformatics. The most widely used is the Tanimoto or Jaccard coefficient.
\[ T = \frac{A \cap B}{(A + B) - A \cap B} \]

where A, B, A&B, are the number of bits set in fingerprint A, B, and A-AND-B.
In a hypothetical example, A, B and A&B are 24, 21, and 19, respectively, resulting in a Tanimoto coefficient of 0.73 (1.00 is perfect similarity). Another way to put it \[ TC = \frac{BC}{B_1 + B_2 - BC} \] (BC=bits set in common, B1 is bits set in 1, B2 is bits set in 2). An example of comparison of analogs of o-chloro-p-aminobenzoic acid.

<table>
<thead>
<tr>
<th>Structure</th>
<th>Tanimoto coefficient</th>
</tr>
</thead>
<tbody>
<tr>
<td>Benzoic acid</td>
<td>0.52</td>
</tr>
<tr>
<td>m-chlorobenzoic acid</td>
<td>0.64</td>
</tr>
<tr>
<td>o-chlorobenzoic acid</td>
<td>0.80</td>
</tr>
<tr>
<td>o-chloro-p-aminobenzoic acid</td>
<td>1.0</td>
</tr>
<tr>
<td>p-aminobenzoic acid</td>
<td>0.70</td>
</tr>
<tr>
<td>p-chlorobenzoic acid</td>
<td>0.66</td>
</tr>
</tbody>
</table>

Sci-Finder has an option to look for similar compounds using a similarity index which (I presume) uses a Tanimoto index of pre-computed bit-strings.

**Similarity using molecular properties.**

There are of course an enormous number of molecular properties
That can be used to compare molecules – some of the more common ones are listed below:

1. **Quantum mechanical descriptors based on the wavefunction (Carbo index)**
   
   \[ C_{ab} = \frac{(ab)}{\sqrt{(a)(b)}} \]


   Molecules............Index.........graph of similarity of pairs

2. **Topological indices (Weiner, Kier and Hall)**

   ![Topological indices](image)

3. **Compute molecular properties:** volume, surface area, logP, pKa, ........vast number – then cluster molecules according to a similarity measure.
We have developed a method to predict the metabolites of molecules using fingerprints and similarity.

**Finding molecules in 3-Dimensions.** So far, we have described the molecular graph and the properties of molecules to find those that are similar. However, molecules can change their properties as they change shape (e.g. their dipole moment, surface area or nmr spectrum). In drug discovery, catalyst design, materials properties etc. the 3D shape of a molecule is critical to its interactions with other molecules – and this effects the properties of the material. Crystals for example may have molecules arranged in different conformations with different interactions with each other (polymorphs), and this has profound effects on their melting points. A biological receptor, typically a protein, which has a “binding site” that is like the lock to the molecular key that interacts with it, typically has exquisite complementarity. This is a view of sialidase (an influenza target) and an inhibitor, and its interactions with the protein (von Itzstein, *Nature Reviews Drug Discovery* 6, 967-974 (December 2007)).
Notice the pattern of hydrogen-bonding and other key interactions. This introduces the concept of pharmacophores, the 3D arrangements of potential interacting regions between molecules and their biological targets.

Here is an example of a ‘pharmacophore’, a 3-D representation of the shape a molecule is thought to take up when binding to a biological receptor: with a protonated amine, the centre of a hydrophobic group (the indole), a hydrogen-bond acceptor (the ketone) and another hydrogen-bond
acceptor (a sulphonamide oxygen). The distances and angles are determined by the 3D shape of the molecule. This was used to design a drug for migraine (Zomig). We can look for similar molecules in our database by using these distance and angle constraints between the fragments shown. To do this:

1. We convert our database to 3D (using e.g. Concord, Corina)
2. We define a “query” using as input the fragments at the key points, distances, angles etc.
3. We match the query to the database, while conformationally searching each molecule of the database (i.e. only those that have the key fragments present).
4. List the results and check they make sense

Here is an example search of our 4.2Million compound database. Indoles were omitted (as in the structure above) to find some novel structures. Here are three examples obtained. These could be tested for their anti-migraine properties. This is an example of what is termed “virtual screening” (screening is the process of testing many molecules against a property or biological target (which is the screen) to find those that satisfy the requirements need e.g. kills a pathogen or cures a migraine).

There are various online programs to assist in this process, here is an example: http://www.click2drug.org/ - Click2Drug contains a number of databases and methods to design drugs.
Using the 3D structure of a protein or other substrate. There are specific instances where we wish to screen libraries of compounds or design specific molecules using the 3-Dimensional structure of a biological target. In this case, usually a protein (but may be DNA, RNA, crystalline materials, substrates like hair or skin) has had its 3D structure determined by x-ray crystallography. We then wish to ‘dock’ the small molecule into the binding site of the larger molecule (like a ship docking in a port). We wish to find the energetically most favourable position, and hence select a molecule from a database for testing, or alternatively design a better one by modification and re-docking.

A database of protein structures (the Protein Data Bank or PDB) contains 94,587 structures. The ZINC database of small molecules contains over 95 million compound structures, many purchasable, in ready-to-dock, 3D formats. This is too many to screen, even on the fastest computers. If we have filtered the database by e.g. molecular weight, favoured chemical structures, availability, chemical stability, cost of purchase etc. we may end up with a few thousand molecules we could dock and score (evaluate their fit to the protein). In this way, we can do “virtual screening” of molecules against a biological target. Since we can create 3D databases of molecules and we have the 3D structure of the protein, it is “simply” a matter of placing the new molecule in the correct position and evaluating the interaction in solution. Above is the anti-cancer molecule Gleevec docked into its primary target. Imatinib is specific for the TK domain in abl (the Abelson proto-oncogene), c-kit and PDGF-R (platelet-derived growth factor receptor). The docked structure is compared with the x-ray structure, and they fit very well. A 3D-pharmacophore can be deduced and used to search our database of 3D structures, then subsequent docking can inform us if the hit molecules fit and these can then be tested.

Virtual screening. When discovering new drugs, catalysts, materials, polymers etc. scientists now have the option to screen molecules experimentally (very expensive) or to screen using computational methods (cheap). Of course, the experiment is best; simply put; the cost of screening e.g. 4.2 million molecules against a single biological target ($20/screen) would be prohibitive even for the largest organisations. Screening such large volumes of compounds and targets is called High Throughput Screening or HTS. So we have to choose a subset of molecules to screen – hence a pre-screen “virtually” in the computer. This is the method applied above to screen for a 3D query. However, there are more options.

Similarity and diversity.

If we have a biological target, and we can only afford to screen 10,000 molecules from our 4.2M molecule database – how do we select a subset to maximise the chances of getting a “hit” – the term used to discover a “lead” structure. There are essentially two ways to do this – similarity and diversity.
Similarity. The first, if we already have data on chemical structures that work, is to find similar molecules to these (e.g. using the methods above). So, we can for example search for all molecules in our database that have a Tanimoto similarity of greater than 0.8 to our list of working molecules (we can vary the Tanimoto index up or down to get the required number of hits from our search), order or make these molecules, and test them. This is an effective way to test the most likely molecules (based on similarity).

Diversity. The second common situation is where we have a biological target and no starting points. In this case, to maximise our chances of getting “hits”, we select from the entire database in a logical way to cover all the chemical diversity of the database (usually computed but may include experimental values too), but with fewer molecules. One way to do this is to select a molecule randomly, eliminate all its “neighbours” having a Tanimoto index greater than 0.8 (for example) then repeat until no molecules remain. All molecules left are dissimilar from each other and span the chemical space. There are many variations on this approach. Commercial screening libraries (e.g. Maybridge HitFinder library (“Selections are made using a clustering algorithm employing standard Daylight Fingerprints with the Tanimoto similarity index clustering at 0.71 similarity”)) of diverse compounds are available.

Combinatorial Chemistry. At this stage, it would be worth mentioning how we can construct virtual libraries of compounds in the computer - and these can also be used for computer-based screening. Of course the selected (virtual) molecules may not be available, and will need to be synthesised. Combinatorial chemistry is as it sounds: we combine reagents and reactions in an array or matrix to generate very large numbers of potential compounds. E.g. for a reaction with R1 and R2 substituents, and 200 of R1 and 200 or R2, we get 4000 possible products. It scales as the product of the R-groups.
Recall the earlier description of SMILES and the related language for reactions SMIRKS. In SMIRKS: CC(=O)O.OCC>[H+].[CL-].CC>CC(=O)OCC is the reaction of an alcohol with a carboxylic acid to make an ester.

A more complex example:

\[ \begin{align*}
\text{[*:1][C:2]=([=O:3])[O:4][H].[*:2][C:5][O:6][H]>[*:1][C:2]([=O:3])[O:6][C:5][*:2].[H][O:4][H]} 
\end{align*} \]

Two sets of acids and alcohol making esters with two combinations of R1 and R2 (*:1 and *:2). So if R1 were 300 and R2 were 300, we would get 9000 products. The SMIRKS for reagent list one and two are fed into the reaction SMIRKS, pattern matched to the amine or acid, and the list of 9000 products generated. We then need to select the ones we really want to make, so could use e.g. molecular weight, number of rotatable bonds, ease of synthesis, price of reagents, similarity to known active molecules etc. and filter out the rest. Then we could apply similarity or diversity selection to create a library of compounds for synthesis which are most likely to work in screening.

The two lectures stop here. However, you may be interested in how (using statistics and machine learning) we can calculate the properties of molecules before synthesis or explain why they act as they do (e.g. as effective drugs).

1. Molecular properties from molecular data

There are many instances where we may wish to predict the properties of molecules or materials. A simple example could be the retention time of a molecule in Thin Layer Chromatography (TLC). If we have a mixture of molecules, it would be useful to know the order in which they would elute. This approach is widely used in Cheminformatics and can be summarised in the following diagram:
This is the most common paradigm for molecular analysis and prediction.

A database of molecules is collected, properties and descriptors of the molecules are measured and computed, and analysis allows the prediction of the properties of as yet unseen molecules. The process is called QSPR (Quantitative Structure Property Relationships) or in pharmaceuticals, QSAR (Quantitative Structure Activity Relationships).

A database may include 20 to 20,000 molecules – depending on the analysis. For example, we may have a database of the structures and melting points of 100 molecules and which to create a model to predict the melting points of similar molecules.

**Molecular data for models.** Molecular data for use in models (descriptors) can be experimental (measured) or calculated. E.g. we could measure the melting point and calculate the solubility. Both can be included in a model. The number of descriptors of molecules we can compute is continuously growing. Examples are : Partial atomic charges, dipole moment, surface area, volume, ovality, number of H-bond donors, number of H-bond acceptors, Principal axes, Homo/Lumo energies, Heats of formation, Polarisability, solvation energies, fingerprints, ClogP, CMR, etc. Descriptors can be scalars e.g. molecular weight), vectors (e.g. dipole moment) or tensors (e.g. principle axes). Some Descriptors are derived from other descriptors (e.g. ovality is derived from surface area and volume), can be derived from the molecular graph (e.g. Keir and Hall descriptors), or be computed using quantum mechanics (e.g. partial electron densities on atoms). A good example of software to calculate these is the freely available PADEL descriptors.

Ideally if we want our model to predict the properties of new molecules, the descriptors should be calculable i.e. we can compute these for the new molecule and not have to measure them.

**Simple approach.** The simplest approach is called “read across”. So if molecule A has a measured property, and molecule B has not had it measured, if molecules A and B are very similar, perhaps they have similar properties and we can predict the property of B. This is a common approach in
predicting the toxicity of molecules. Or another example here. E.g. Sarin (a nerve agent) has an organophosphorous group, if we are going to make some “similar” analogs (e.g. Tanimoto > 0.85) of this with organosphorous functional groups for e.g. anti-cancer testing, we need to be careful!

**Picture the data.** The next simplest method is to plot the data. Always the best if possible. Here is a real example, two descriptors CMR (the size of the molecule) and logD (the distribution coefficient between octanol and water) are calculated, plotted and annotated with their ability to be absorbed in the intestine. The white areas are absorbed, the shaded molecules are not. So as drugs, they would not be orally absorbed therefore useless in pills. The analysis of data of this type has led to a very important advance in designing absorbed drugs called Lipinski’s Rule of 5. For an orally absorbed drug the guidelines are deduced from data to be:

- Not more than 5 hydrogen bond donors (nitrogen or oxygen atoms with one or more hydrogen atoms)
- Not more than 10 hydrogen bond acceptors (nitrogen or oxygen atoms)
- A molecular mass less than 500 daltons
- An octanol-water partition coefficient, log P, not greater than 5

Nice simple rules derived from a complex dataset.

**Preparing the data.** Before creating models, we should say something about data preparation as this will influence the quality of the models produced. With data on series of similar chemical compounds, there are often two main problems. A major problem is data scaling – i.e. how important is one descriptor compared to another descriptor? e.g. a volume calculation on our series of molecules could vary from 25-1000 Angstroms\(^3\) while an atom excess charge may only vary from +0.3 to –0.3 electrons over the same series. A simple way to alleviate this problem is autoscaling. Each descriptor variable is given a mean of zero and a standard deviation on one.

$$x'_i = \frac{x_i - \langle x \rangle}{\sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \langle x \rangle)^2}}$$

of zero and a standard deviation on one.

Additionally, when many descriptors for molecules are calculated, there is a common situation where the variance in the descriptors and the distribution of the measured values is very similar. (note: variance is a measure of how far a set of numbers is spread out and is the average of the squared differences from the Mean). This is ‘inter parameter correlation’. This can give undue combined influence for similar descriptors. A simple way around this is to eliminate one of the descriptors e.g. surface area of a molecule and its volume are often highly correlated

The correlation coefficient is calculated from the least squares calculation (see later) between the descriptors (Pearson’s correlation coefficient).

$$r = \frac{\sum (x_i - \langle x \rangle)(y_i - \langle y \rangle)}{\sqrt{\sum (x_i - \langle x \rangle)^2 \sum (y_i - \langle y \rangle)^2}}$$

Values of greater than 0.9 usually imply that one of the parameters may be omitted.
Statistics, and pattern recognition. A more complex approach is to construct a Model from the data and use this to predict properties of molecules. This involves molecules, their data, a training set of molecules and the relevant molecular descriptors.

We can construct a statistical model and use this to predict properties of as yet un-synthesised molecules. In its simplest form:

1. The first step is constructing a database of molecules. These can be stored in a database, as above.
2. Calculate “molecular descriptors”. These are such things as atomic weight, surface area, dipole moment...there are thousands of these that are accessible.
3. Put these in a table – here is an example of some drugs with variable measured biological activities (MES – a measure of anti-epileptic activity) and three descriptors calculated for each: the log_{10} of the molecular weight, the octanol/water partition coefficient, and the dipole moment).
4. Using a model building method (statistics) relate the descriptors to the property we wish to predict.

The types of model are traditionally divided into supervised and unsupervised methods. A supervised approach uses data to which a model is fitted. In unsupervised methods, the data is classified from the descriptors, without creating a model.

**Supervised methods.** The most common method is linear regression. Simple linear regression fits a straight line through the set of n points in such a way that makes the sum of the squared residuals of the model (that is, vertical distances between the points of the data set and the fitted line) as small as possible. The equation we obtain can be used to predict a new property based on the descriptors calculated or measured for the new molecule. This is from Wikipedia with some annotation:

\[
\text{Find } \min_{\alpha,\beta} Q(\alpha, \beta), \text{ where } Q(\alpha, \beta) = \sum_{i=1}^{n} \varepsilon_i^2 = \sum_{i=1}^{n} (y_i - \alpha - \beta x_i)^2
\]

Q is the function we want to obtain and minimise
Alpha is the correction factor to move all the points so the line goes through the origin
Beta is the coefficient to multiply our descriptor by.
Epsilon is a residual
it can be shown that the values of \( \alpha \) and \( \beta \) that minimize the objective function \( Q \) are
where \( r_{xy} \) is the sample correlation coefficient between \( x \) and \( y \), \( s_x \) is the standard deviation (the standard deviation is found by taking the square root of the average of the squared differences of the values from their average value) of \( x \), and \( s_y \) is correspondingly the standard deviation of \( y \). A horizontal bar over a variable means the sample average of that variable. For example: \( \bar{xy} = \frac{1}{n} \sum_{i=1}^{n} x_i y_i \).

Substituting the above expressions for \( \hat{\beta} \) and \( \hat{\alpha} \) into the original equation

\[ y = \hat{\alpha} + \hat{\beta}x, \]

So if our new molecule has a descriptor \( x \) we can now calculate the property \( Y \).

\( r_{xy} \) is called the correlation coefficient and varies between zero (no fit) and one (perfect fit).

One well known example using measured and calculated descriptors to predict a property is the Yalkowski equation, to predict (pred) aqueous solubility and compared it to experimentally observed values (obs) – we need to know the melting point and the calculated logP (logP is the octanol/water partition coefficient which is easily calculated).

\[
\log Sw = -0.01 (MP-25) - \log Kow + 0.5
\]

Specific examples shown below, and the equation applied to \( n=949 \) compounds (\( r^2=0.74 \)).

(Multiple Linear Regression. Regression can be over many variables, not just one. This is multiple linear regression, and is the most commonly used method to create QSAR/QSPR models (and its related method step-wise regression). Here is an example of a multiple linear regression equation to predict solubility of drug-like molecules.

\[
\log S = 3.866 - 0.0194 \text{SASA} - 0.514 \text{HBAC} + 0.578 \text{HBDN} + 1.343 \# \text{Amine} + 1.224 \# \text{Amide} - 116 \text{HBAC} * \text{HBDN}^{1/2} / \text{SASA} + 0/182\# \text{rotor} - 0.00405 \text{WPSA}. \]

The descriptors and the fit to the data is shown below.

Are the models any good? You may have noticed a serious flaw in all of this. We haven’t checked how good the methods actually predict, just how well we can fit the data. To do this, we need to validate the model.

The easiest way is to split the data into two parts – a **training set** and a **test set**. So, e.g. we may have measured the solubility of some simple amides and we now wish to predict the solubility of as yet un-synthesised amides and only make those which are predicted to be soluble. To test if the model is any good, we train on a **training set** (e.g. 75% of the data randomly selected), test on a test set (e.g. the 25% of the data remaining). If we are rigorous, we can then evaluate how good the model seems to be on a completely **unseen test set**. We could then plot the measured vs. predicted values and look at the fit. Over-fitting is a major problem in all of these models so much work has gone into validation.

A more complex approach is to use **cross-validation**. In Cross-Validation, a training and test set are generated randomly N-times. The usual split is in the e.g. 70/30 ratio. The model is generated from the training set and used to predict the test set. Over the N-runs, the mean of the sum of squares of the errors in the predictions are accumulated and scaled such that the percentage error as a proportion of the variance in the data is reported – usually called the PRESS value (Predictive Residual Sum of Squares).

\[ \text{PRESS} = \frac{1}{n} \sum_{i=1}^{n} (\hat{y}(i) - y)^2 \]

A value for the cross validated \( R^2 \) varies between -1 and one. Less that 0 indicates no predictive ability up to complete model fitting (1). A cross validated \( R^2 \) better than 0.5 implies that there is some prediction in the model. (Don’t confuse this with the regression \( r^2 \)). The dataset of course has to be large enough to provide sufficiently large sub-samples of the data for training and testing. There are many variations on this including bootstrapping, repeated random sub-sampling, y-scrambling. **Overfitting** of data in QSAR is discussed here.

It is always informative to plot the model relationships. These are different datasets with \( r^2 \) values. Notice particularly that some of the data would be better fitted with a non-linear model, some data

---

LogS is the computed solubility.
SASA, solvent accessible surface area
WPSA term is the surface area for all halogens, sulfur, and phosphorous atoms.
No of amines #amine, carboxylic acids #acid and amides #amide
number of rotatable bonds #rotor
HBAC is hydrogen bond acceptors and HBDN is hydrogen bond donors
is clearly related and would be better modelled as subsets,

The regression equation should include statistical tests of robustness such as an F-test, number of points, standard deviation, correlation coefficient e.g.

\[
\log 1/C = 1.15 \pm 0.2 \sigma + 1.46 \pm 0.4 \sigma^2 + 7.82 \pm 0.2
\]

Also, each of the parameters (the numbers multiplying the descriptors) should have the estimated error. If this error is bigger than the multiplier, the descriptor is not significant and should be removed. Many publications don’t include this. The descriptors used should have some scientific rational for their inclusion – they can tell you how the model relates to the real phenomenon being modelled - many times descriptors are included just because they fit the model, not because they make any kind of sense (e.g. the day the molecule was synthesised).

Importantly, the best model may not be linear. In this case, we have introduces squared terms in to the equation to accunt for the parabolic nature of the data we wish to fit. e.g. \( P = ax^2 + ax + by^2 + by + C \)
Unsupervised methods. In the previous examples, data was fitted to a model, usually predicting a numeric value of the desired property. However, it is also possible to cluster the data, and hence make predictions about a particular class a new molecule will fall into e.g. is it toxic or non-toxic. This is “guilt by association”. The most common approach to do this is cluster analysis, (reference paper) which includes a diverse set of approaches. Defining what a cluster is, is often difficult, and may be subjective. Hierarchical clustering and k-means clustering are common approaches. Clustering involves finding the distance between all points of the data (e.g. the Tanimoto distance) usually using the Euclidean distance or the Manhattan distance. The clusters are then determined by either a bottom-up approach (agglomerative) or by a Divisive approach (top-down). This is an example of what is termed a “greedy algorithm”, where we look for increasing links to other cluster as we get to more similar molecules. The results are usually displayed as a dendrogram. There are many examples of clustering in Cheminformatics e.g. clustering chemical scaffolds, blood-brain barrier penetration, clustering environmental pollutants,— and different ways of visualising clusters. Whether a dataset can be clustered into different classes depends on the degree of separation of the classes, which of course depends on the descriptors used to describe the molecules, the clustering method, and how the clusters are identified as being in separate classes e.g. toxic or non-toxic may work, but soluble and insoluble may not. Here is an example of Clustering 60 human cancer cell lines based on the relative biological activity patterns of 25,023 molecules.
PCA. Another commonly used unsupervised method for data clustering is Principal Components Analysis. This is a method to reduce the dimensionality of the data — what this means is that if, for example, if we have 100 descriptors, which are too many to be visualised, we can reduce the number of dimensions to only two, and still encapsulate the influence of the descriptors on the model we use to visualise the data. PCA is the transformation of a set of correlated variables to a set of orthogonal uncorrelated variables called principal components. These new variables are a linear combination of the original variables in decreasing order of importance. The PC’s each maximise the variance in the data in orthogonal directions and are ordered by size. Usually only a few components are needed to explain (>90%) of the variance in the data — or the descriptors are not relevant. This is shown pictorially above. PC1 here explains most of the variance in the data, so we could probably replace X and Y (2 variables) coordinates by PC1. The objective of PCA is to find the eigenvalues (how much of the variation of the data is explained by each component) and the components themselves (the factors and their loadings, which are the equations used to describe the vectors that are spanning the maximum variation in the data in orthogonal directions).
This is exactly the same method that is used in quantum mechanics to obtain molecular orbitals from a linear combination of atomic orbitals. A step-by-step tutorial if you are interested is here. So this basically investigates the “variance” in the data. From this we can: find out which descriptors are probably important, how many of these equations (eigenvectors) we need to explain a lot of the variance (the eigenvalues). We can then plot and colour the plots to see if there is a classification of the property we are interested in. Here is an example of PCA combined with multiple linear regression to predict solubility. Solubilities were measured for 396 solutes in 153 solvents, many descriptors calculated, and the resulting models derived from PCA combined with multiple-Linear-Regression. Here is a slightly more complex example, in which we investigate the concept of “chemical space”, that is how similar molecules cluster together based on computed descriptors and a similarity index and how subsets of these are clustered in a large universe of compounds with objective of predicting melting points.

\[ Y_{ik} = \bar{Y}_i + \sum_{p=1}^{p} b_{ip} x_{pk} + \epsilon_{ik} \]

- **Y**<sub>ik</sub> = **Y**<sub>i</sub> + \sum<sub>p=1</sub>^{p} b_{ip} x_{pk} + \epsilon_{ik}
- **Y** = data matrix
- **b** = loadings (measure of the variation between variables)
- **x** = scores (measure of the variation between samples)
- **\epsilon** = eigenvalue

**Figure 3.** First two principal components of the principal MDPI data set (gray), compared to the external validation drug data set (black) in 2D descriptor space. The first two principal axes explain 32.61% and 12.81% of the variance, respectively. Owing to the phyicochemical diversity of the principal data set (gray) it comprises a much larger area in chemical space than the, comparably, more homogeneous drug data set.

The 2D descriptors include computed descriptors (such as charge, van der Waals volume, and molecular refractivity), subdivided surface areas (atomic contributions to logP and molecular refractivity), counts of elemental atom types and of bond types, Kier/Hall connectivity and kappa shape indices, topological indices (Wiener index and Balaban index), pharmacophore feature counts (number of acidic and basic groups and hydrogen bond donors and acceptors), and partial charge descriptors. The conformation-independent 3D descriptors include potential energy terms (such as total potential energy and contributions of angle bend, electrostatic, out-of-plane, solvation, etc. terms) and surface area, volume, and shape descriptors (among them water accessible surface area,
mass density, and principal moments of inertia). So this is basically describing a series of molecules in many ways, then compressing the plot into two dimensions so we can look for patterns in the properties of the molecules.

**Machine Learning.** Much effort has gone into new methods for classification and model building which are based on heuristics rather than statistical variance. This is a branch of artificial intelligence research and the objective is to create methods that can learn from data, and may include additional rules or guidelines that help form a predictive or classification model. One of the first methods was **Rule Induction.** An algorithm termed **ID3** formed the **basis** of a number of rule based systems (also termed Expert Systems) to predict chemical synthesis, toxicity or **carcinogenesis.** These methods created decision trees. In essence, the program is presented with a number of cases, and then rules are generated from these. Here is an **early example** of a rule-tree which predicted biological activity at a specific receptor for a specific class of compounds. To get a prediction, calculate the descriptors for the molecule and follow down the tree to get a classification.

An early example was **LHASA-** which predicted chemical synthesis plans from a library of reactions. A good example is **Derek** – a program that uses Rules derived from known experimental results to predict toxicity of molecules. The molecule query is broken down into small fragments which are individually assessed for toxicity. The rules are then combined to give an overall assessment. A recent development of this approach is termed **Random Forest**, as many rule-trees are derived and combined to form a more predictive rule set. This is the most popular methods along with **Support Vector Machines.**

Another example of machine learning methods is **neural networks.** These are trained on a number of cases, as before, but in this case the architecture of a neuron is incorporated into a network of connected elements. Here, a number of inputs (descriptors of molecules for example) are input, the connections between the input layer, hidden layer and output are adjusted, and the model created can be trained to produce the correct results. **Here is an example** using chemical compounds, and a **book.** The example below **predicts TLC retention times.** It uses the partial charge on nitrogen and oxygen atoms, the logP, dipole moment, electron affinity, CMR (molecular size), and polarisability.
More recent methods are Inductive Logic Programming, Support Vector Machines, Bayesian Networks etc. Many of the more mathematically based approaches are devised to create classification surfaces between sets of points in a n-dimensional hyperspace. The classes obtained can then be projected onto a two-dimensional surface. An example is a non-linear map, (Sammon mapping) which projects data having n-dimensions onto a two-dimensional plot while attempting to maintain the relative inter-point distances. So if we have 50 molecules, with 20 descriptors (each is a different dimension) calculated for each molecule, we can project their descriptors onto a 2-dimensional plot and see if they cluster in some interesting way.

More recent methods – the explosion of interest in AI methods. The recent progress in Deep Learning, applied to many cheminformatics problems, is augmenting traditional statistical and machine learning approaches. An example of prediction of solubility is shown here. Deep learning typically uses neural networks of complex architectures to create a model of the data. Autoencoders produce a deep representation of the data that encapsulates the variation in the data with a reduced dimensionality. Typically very large datasets are most successful. An example from our work here uses deep neural networks to represent the chemistry of a cancer tumour using mass spectrometry.

Validation of Models. Again, we must test if the models are valid. We can use training and test sets or cross-validation as before, to test for the stability and predictivity of the models. There are two additional approaches that we should mention. They are used for classification models. We define the following:

True positive: the correct answer for being a member of a class (e.g. is toxic)
True negative: the correct answer for not being a member of a class (e.g. is non-toxic)
False positive: The wrong prediction e.g. is predicted positive, but is in fact negative
False negative: The wrong prediction e.g. is predicted negative, but is in fact positive

Now we can use a Mathews Correlation Coefficient to test the robustness of the model:

$$\text{MCC} = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$
If any of the four sums in the denominator is zero, the denominator can be arbitrarily set to one. A larger value indicates increased confidence in the model. From this, have also been defined:

\[
\text{Precision} = \frac{tp}{tp + fp} \quad \text{True negative rate} = \frac{tn}{tn + fp}
\]

\[
\text{Recall} = \frac{tp}{tp + fn} \quad \text{Accuracy} = \frac{tp + tn}{tp + tn + fp + fn}
\]

Precision is the probability that a (randomly selected) retrieved example is relevant. Recall is the probability that a (randomly selected) relevant example is retrieved in a search. From this, another number can be generated, the F1 measure:

\[
F = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}
\]

Similar to this approach is the **ROC curve** (Receiver Operator Characteristics). This was initially developed to classify radar returns. It uses the same definitions of TP/TN/FP/FN and incorporates these into a graph where the y-axis is the fraction obtained that are correct, while the x-axis is the fraction of the data examined to find that correct amount. (It is created by plotting the fraction of true positives out of the positives (TPR = true positive rate) vs. the fraction of false positives out of the negatives (FPR = false positive rate), at various threshold settings. TPR is also known as sensitivity (also called recall in some fields), and FPR is one minus the specificity or true negative rate.). Below is a ROC curve.

![ROC curve](image)

The blue line is a model prediction and we see that we have ranked 80% of the true positives in only 10% of the sample, so the enrichment is good. If we were screening for a new biologically active molecule using this model, we could select only 10% of the molecules to screen (at greatly reduced cost) and still have an excellent chance of finding active molecules. The diagonal line means no model, above the line is better enrichment, below is worse than random selection.

**Putting Cheminformatics processes together.** All of these methodologies could be very cumbersome to use – different software, file formats, conventions. There has been significant effort in designing and developing **pipeline** software. That is having a series of components that can easily be assembled (graphically) into a process. Examples are [PipelinePilot](https://pipelinepilot.com), [Knime](https://www.knime.org), [Taverna](https://www.taverna.org). Here is an example showing linear and branched processes:
ELN’s. Another important development in Cheminformatics (and indeed in all science data) is in data capture – using Electronic Lab Notebooks: to input experiments, capture data from instruments and to digitally sign the data (time-stamped so it can be patented with that date). Many large organisations have moved fully over to ELN technology to archive, make searchable and store knowledge on their valuable data. In the chemistry department we use the IDBS notebook. There are many others available.

Robert C. Glen (October 2018)