Spinifex

Spinifex

Advanced chemical similarity software using a graph-based Maximum Overlapping Set (MOS) approach

Desert Scientific Software

Molecular Similarity

'Science involves discovering the similarities between things that are different and the differences between things that are similar.'

A. Schopenhauer

There are distinct approaches in the field of molecular similarity:

1D Representations - the very language of chemistry;

2D Descriptors - focus is on chemical structure and synthesis; and

3D Molecular Fields - pharmacophores embrace space of small molecule biological activity

Chemical Structure Similarity

There are many approaches to chemical structure similarity:

- Physiochemical parameters/QSAR - Hansch
- Fingerprints - MDL keys, Daylight
- Graph matching - Willett
- Atom types - Sybyl, Ghoose/Crippen
- Connectivity indices - Hall

For pairs of molecules, there are many ways to generate, and interpret, similarity coefficients

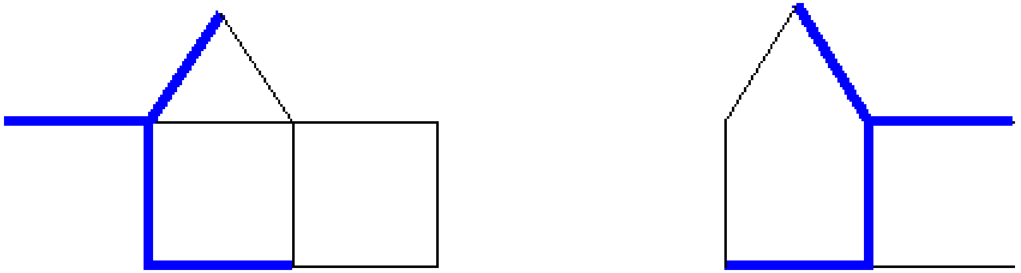
For sets of molecules, there are many ways to do cluster analysis

Chemical Graph Matching Using Spinifex

Developed in collaboration with F.Hoffmann-La Roche

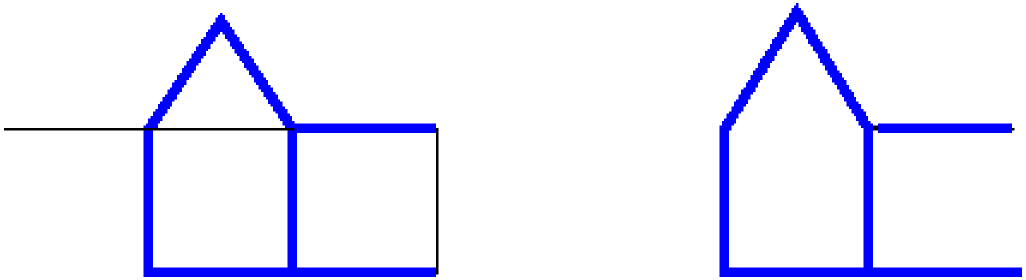
- based upon Willett approach - Raymond et. al., 'Heuristics for Similarity Searching of Chemical Graphs Using a Maximum Common Edge Subgraph Algorithm', JCICS, 2002
- generates Maximum Overlapping Set (MOS) using clique detection algorithms (in an edge induced correspondence graph) and chemical heuristics (for minimising search space)
- the MOS is a generalisation of the Maximum Common Subgraph (MCS) and can be thought of as the largest set of substructures that two compounds share

What is a MOS?



MCS – Maximum Common Subgraph

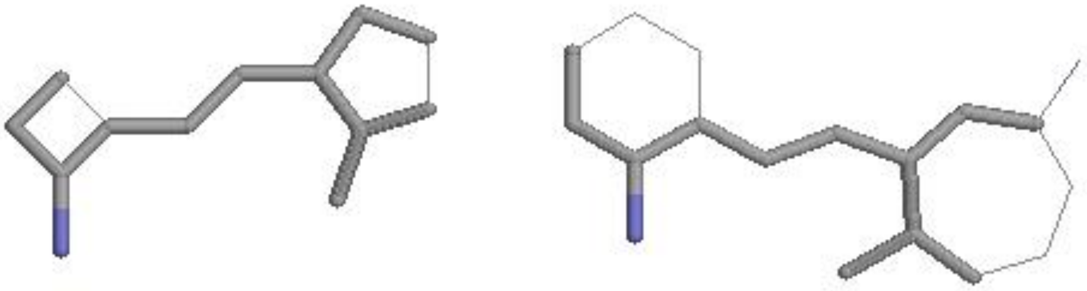
– maximal isomorphism between two node–induced subgraphs



MOS – Maximum Overlapping Set

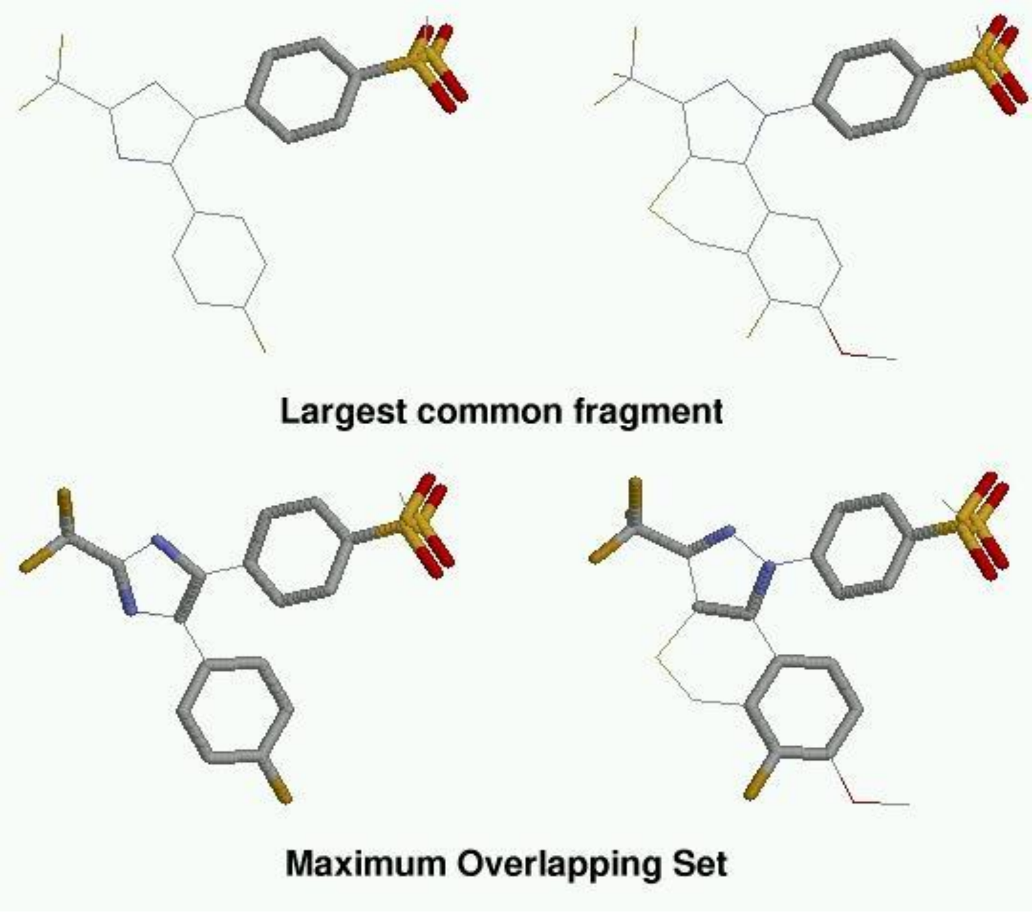
– maximal isomorphism between two edge–induced subgraphs

A Simple Example



An Example with Unconnected Fragments

The MOS has much more information than the largest common fragment for two COX2 inhibitors

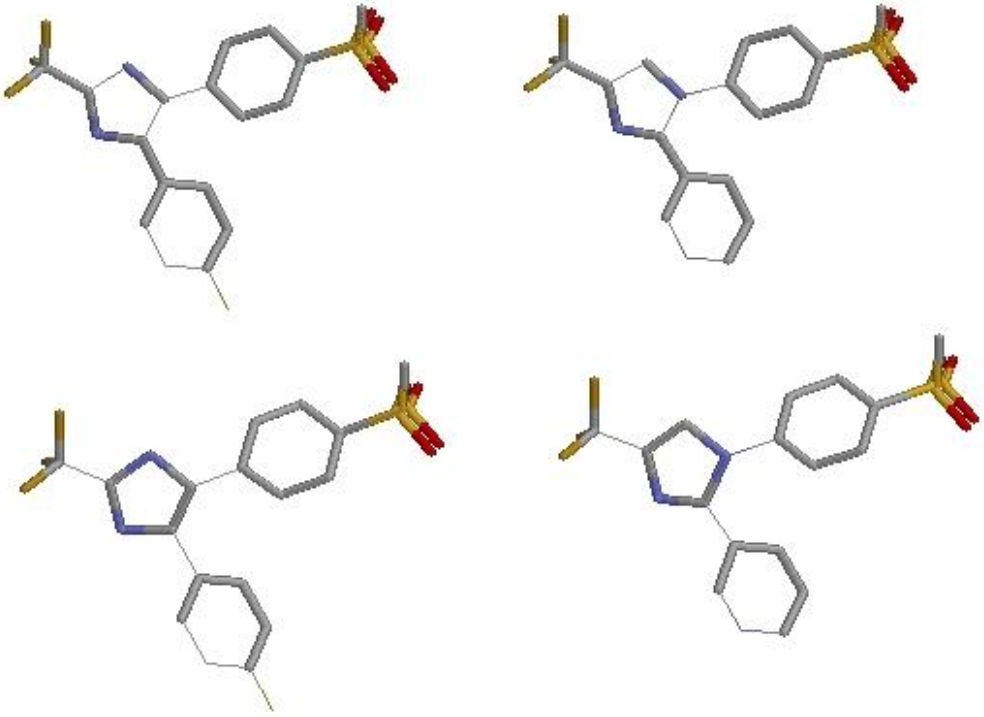


Largest common fragment

Maximum Overlapping Set

Graph Similarity Getting More Complicated

For two COX2 inhibitors, the better solution is not necessarily the one with the largest common fragment



For any pair of molecules, there can multiple solutions to the MOS!

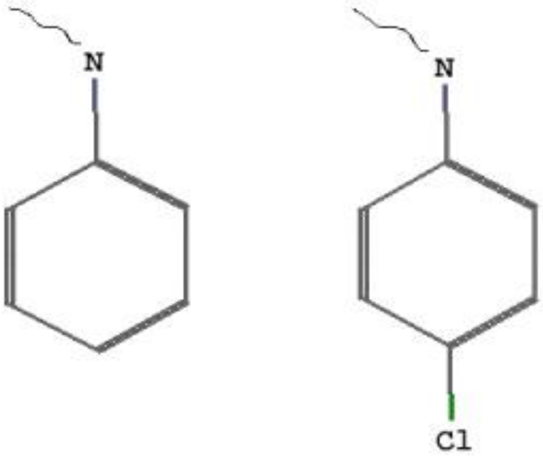
How Spinifex Works

1. Read input SDFs using Python OELib/OEChem - delete hydrogens, determine aromaticity, atom types, bond types, ring properties
2. Compute Smallest Set of Smallest Rings (SSSR) using Figueras approach
3. Identify redundant edge matchings for rings
4. Identify redundant edge matchings for hydrocarbon chains
5. Identify triangle-trinode inequalities
6. Create edge graphs for input molecules
7. Build correspondence graph, handling 5 to 7
8. Find maximum clique in correspondence graph
9. Compute similarity coefficient or create SDF solutions from input molecules
10. Clustering
11. Visualise results

Redundant Edges

There are 12 ways that two benzene rings can be mapped together

In the example below, only 1 mapping need be considered



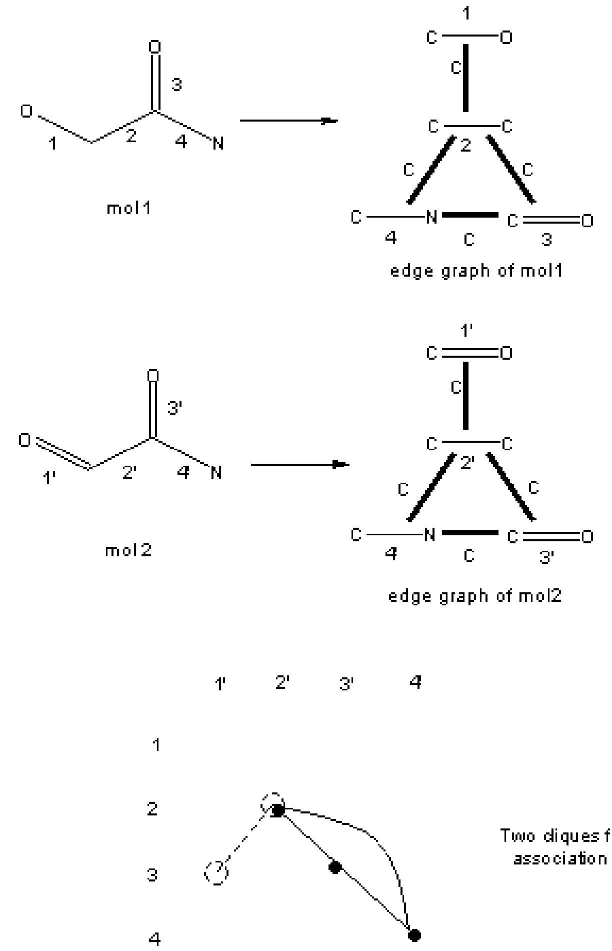
Rings are handled in the following order - benzene, hydrocarbons, aromatic heterocycle rings, then all others.

This means that a benzene ring will more likely match another benzene ring rather than partially match an aromatic heterocycle in the solutions

Ring handling heuristics have DRAMATIC impact on performance

How to Build an Edge Graph

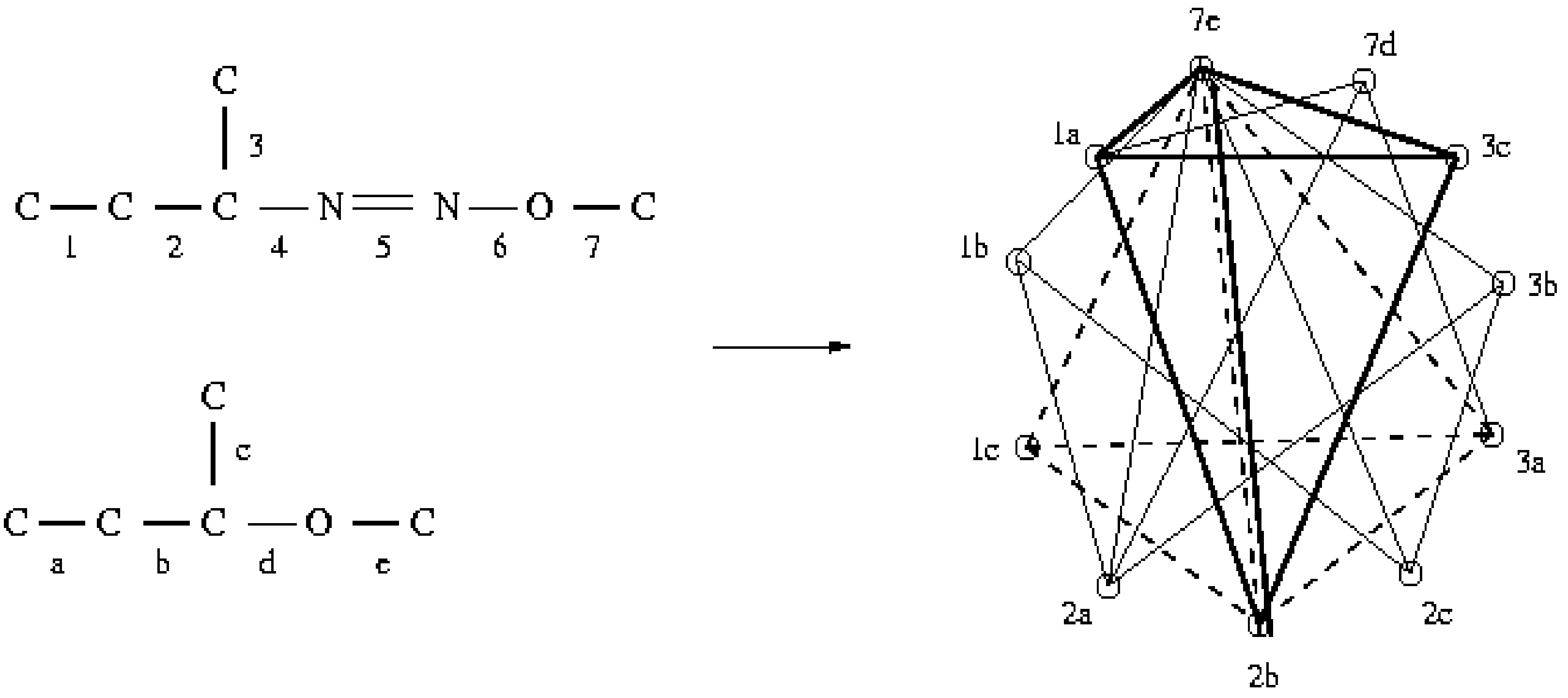
from Raymond et. al, JCICS, 2002



Two cliques from the association graph

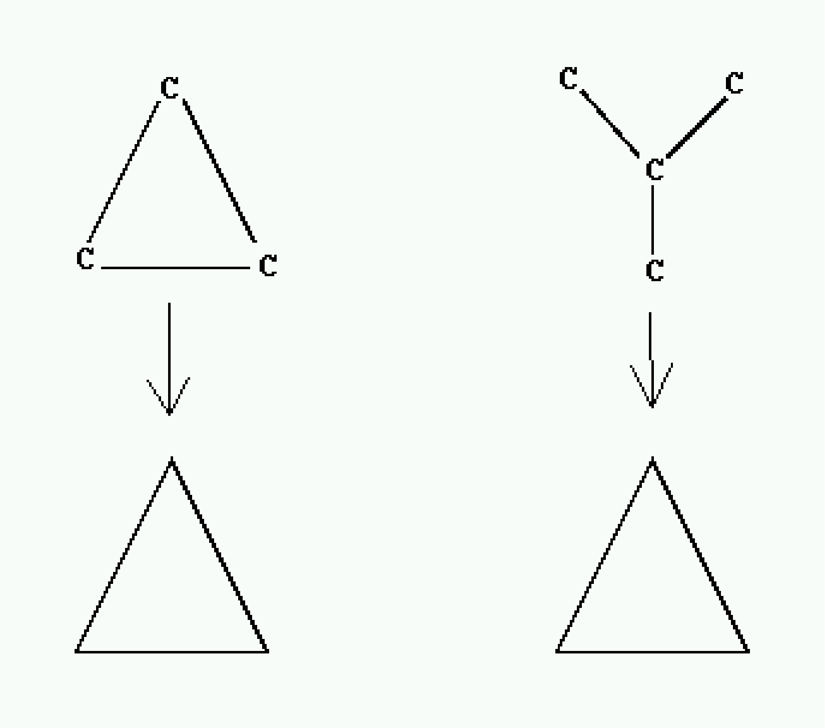
How to Build an Edge Graph (cont.)

from C.K. Chen, University of Hawaii



Triangle-Trinode Inequalities

Both cyclopropyl and isobutyl have identical edge graphs but we don't want them to match



Edge and Vertex Weights

Default method - molecules compared as graphs weighted according to atomic number (2-120), with nitrogens types differentiated and halides grouped together, and according to bond order (1-4)

Molecular graphs can have alternative weightings: using Sybyl atom types; using pharmacophore descriptors (HB donor/acceptor, VDW aliphatic/aromatic)

Spinifex can compare molecules as featureless graphs - all carbon atoms and/or all single bonds (slower computations)

Atom 'colouring' can also be read from sdf data block

Spinifex

Clique Detection Algorithms

Subgraph isomorphism problems have unknown complexity - algorithms tend to be exponentially hard

For clique detection, many algorithms have been tested in Spinifex

- Algorithm 457 (Bron and Kerbosch, 1973)
- rambin (http://www.twisted-helices.com/computing/rambin/rambin.htm, 1995)
- dfmax and nmclique (ftp://dimacs.rutgers.edu/pub/challenge/graph/solvers, 1993)
- wood (Wood, Oper. Res. Lett., 1997)
- pardalos3 (Pardalos et. al., J. Glob. Opt., 1992)
- rascal (Raymond et. al., Comput. J., 2002)

Currently use rascal algorithm for clique detection

The **REAL** speed-up comes from pruning search tree before-hand

Similarity Metric

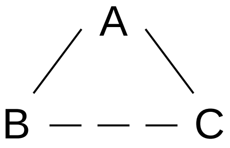
from Raymond et. al, JCICS, 2002

$$\frac{(\ \text{NumVert(MOS)} + \text{NumEdge(MOS)}\ )\text{**}2}{\text{(NumVert(G)+NumEdge(G))*(NumVert(H)+NumEdge(H))}}$$

Similarity Metric with Penalty Factor

A penalty factor can be used to reduce the similarity metric if the arrangement of the three largest fragments in the MOS are not the same for a pair of molecules:

Type 1:



A

B — — — C

Type 2:
A-B-C

Type 3:
B-A-C

Type 4:
A-C-B

Also account for two molecules that share a large common fragment:

AAAAAAA-B considered similar to AAAAAAA-CCCCC

It is possible to regenerate similarity metrics using alternative values for the penalty factors without re-computing chemical similarities.

Spinifex Command Line Options

```
        spinifex.py - solve maximum common substructure problems

    SYNOPSIS:

        spinifex.py [OPTIONS] sdfile1 [sdfile2]

        Default mode compares first mol in sdfile1 with first mol in sdfile2
        and writes sdf of the MOS from atoms and coords from sdfile1

     OPTIONS:

        -a              write both sets of atom numbers of the MOS
        -b              write both sets of bond numbers of the MOS
        -c              run all vs all comparisons. See Note 1
        -d              disable atom typing by sybyl types and H counting. See Note 2
        -e              write both sets of atom numbers of the largest fragment
        -f   'int‘      use graph with reduced features. See Note 3
        -g              graphically display highlighted MOS for both structures
        -h              write this message
        -H              write this message and additional Notes
        -m              compare first mol in sdfile1 with all mols in sdfile2
        -n              write sdf of largest fragment from the maximum clique
        -o              write sdf of the MOS from atoms and coords from sdfile2
        -q              overlay molecules by RMS and create a transformed sdfile2
        -q   -g         overlay molecules by RMS and graphically display results
        -r   'int'      reset timeout. See Note 4
        -s   'int'      write similarity metric. See Note 5
        -S              'a,b,c' penalty factors for similarity metric 2. See Note 5
        -t              write timings (use only with -m and -n options)
        -u              show timings and any similarity metric penalty for each comparison
        -v               verbose output to stderr
```

Parallel Spinifex

- Spinifex can run in parallel on a cluster of linux machines
- Working with sets of molecules belongs to the class of problems termed "embarrassingly" parallel
- PVM, Parallel Virtual Machine (www.epm.ornl/pvm/), wrapped by a python extension module, pypvm.py (W. Michael Petullo, wp0002@drake.edu), is used for inter-process communication
- Parallel Spinifex splits a large job into multiple, small tasks
- Computations done using master/slave paradigm (also known as host/node or crowd computation
- The speed of a job increases almost linearly with the number of available CPUs
- The application used to launch a parallel Spinifex job is spinifex_pvm.py
- XPVM can easily be used to monitor jobs and tasks
- 3000 HTS hits can be compared, all versus all, overnight

Clustering

Clustering methods that have been tested include Jarvis-Patrick, modified Jarvis-Patrick using exclusion spheres, and hierarchical methods such as complete linkage and UPGMA

Best results obtained using sequential agglomerative hierarchical clustering method UPGMA (Unweighted Pair Group Mean Average) - provides results similar to a chemist's view

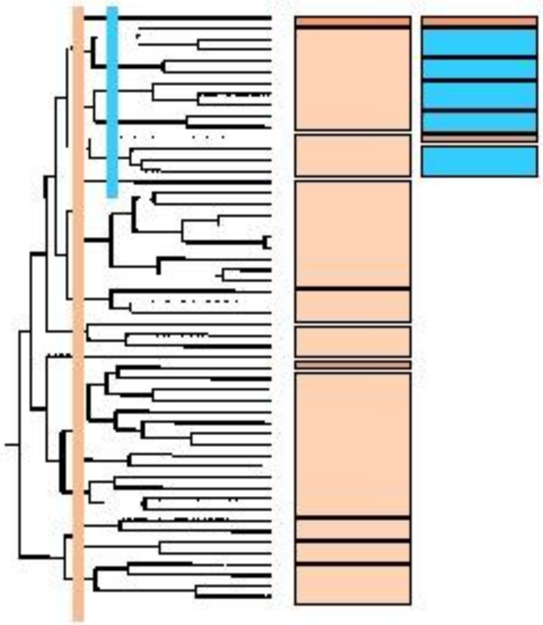
The UPGMA algorithm generally isolates true singletons and produces relatively few coherent clusters not containing outliers

Many clusters can be quickly analysed using alternative commercial tools such as Spotfire

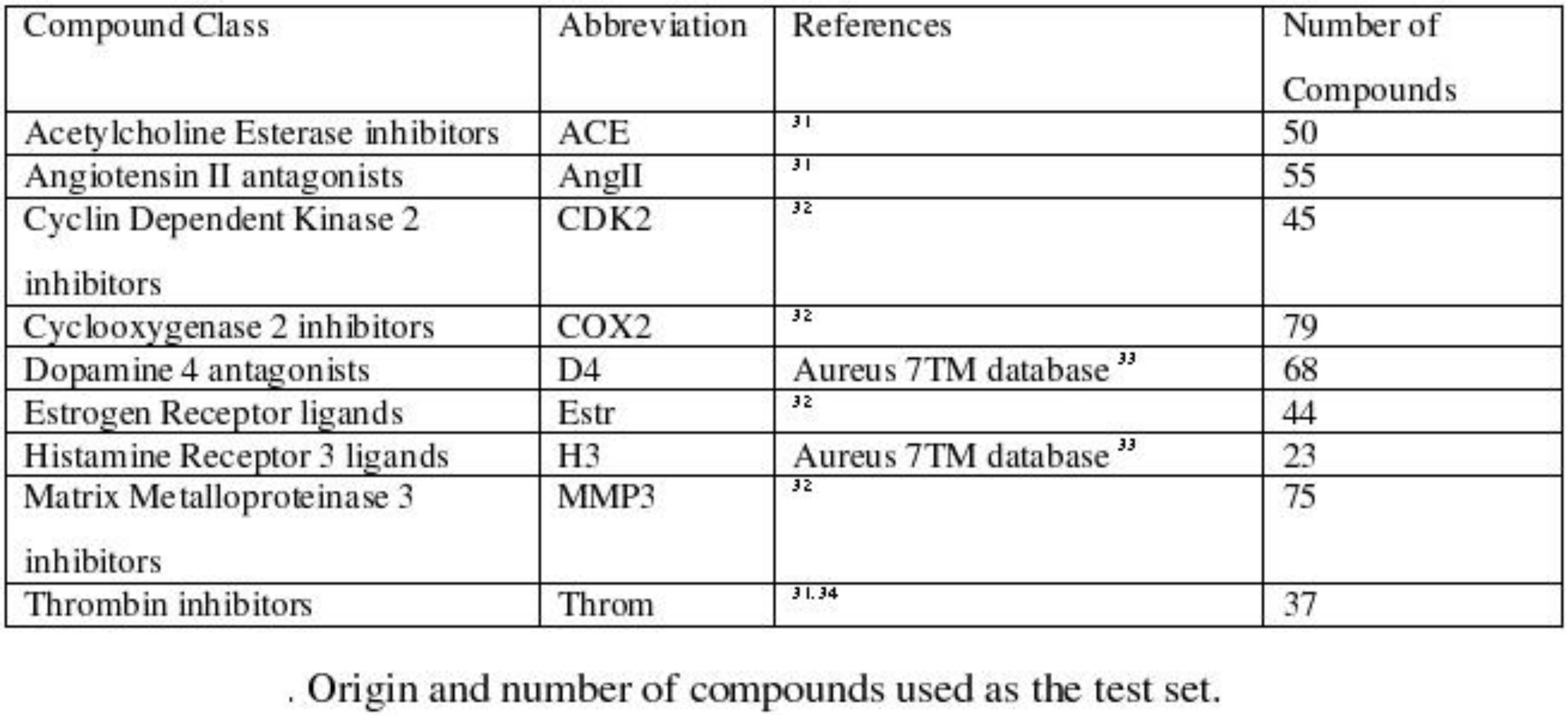
Nearest neighbours for any cluster can be easily identified when using hierarchical clustering

Clustering (cont.)

A hierarchical tree can be cut into clusters at a number of similarity levels and individual clusters can be viewed using Spinifex utilities and Rasmol



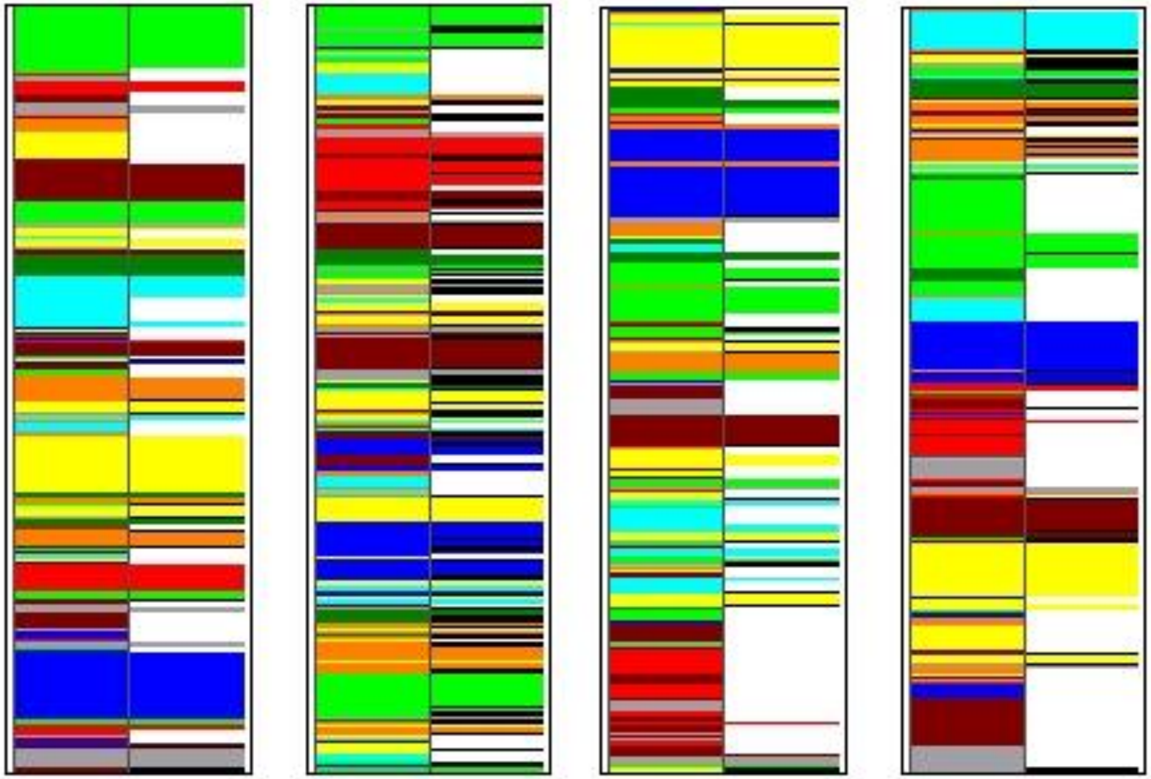
Clustering Data



Compound Class	Abbreviation	References	Number of Compounds
Acetylcholine Esterase inhibitors	ACE	[31]	50
Angiotensin II antagonists	AngII	[31]	55
Cyclin Dependent Kinase 2 inhibitors	CDK2	[32]	45
Cyclooxygenase 2 inhibitors	COX2	[32]	79
Dopamine 4 antagonists	D4	Aureus 7TM database [33]	68
Estrogen Receptor ligands	Estr	[32]	44
Histamine Receptor 3 ligands	H3	Aureus 7TM database [33]	23
Matrix Metalloproteinase 3 inhibitors	MMP3	[32]	75
Thrombin inhibitors	Throm	[31,34]	37

. Origin and number of compounds used as the test set.

Clustering Results



(a)
MOS

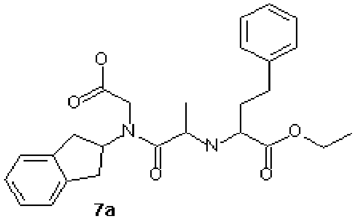
(b)
MCS

(c)
Daylight

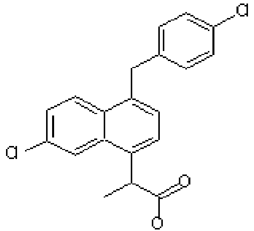
(e)
MACCS

Clustering Results (cont.)

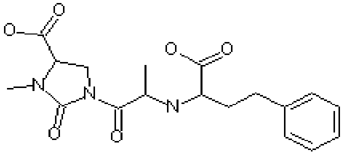
- Scaffolds not necessarily conserved within clusters - pharmacophores often conserved
- ACE inhibitors (7) and COX2 inhibitors (8) are shown below
- Common pharmacophore features are shared
- Only the MOS method is able to cluster these together



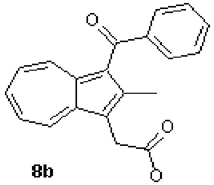
7a



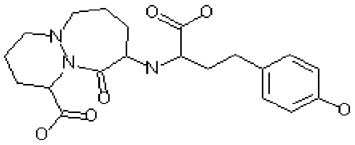
8a



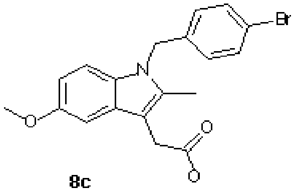
7b



8b



7c



8c