

Hierarchical clustering of chemical structures by maximum common substructures

Miklos Vargyas



ChemAxon
•Solutions for Cheminformatics

Clustering made human

Miklos Vargyas



ChemAxon
•Solutions for Cheminformatics

Why do we cluster?

to reduce the number of objects to deal with

- group subsets together
- represent each group by one member of it

Conventional approaches

Similarity based

- fingerprints or other molecular descriptors
- high dimensional artificial abstract chemical spaces
- similarity measures

Jarvis-Patrick clustering

```
jarp -i SC1000.cfp -m 0 -f 1024 -t 0.6 -c 0.1  
-y -z -o SC1000.jarp.t0.6.c0.1 -g
```

Number of objects = 999

Number of clusters (without singletons) = 2

Number of singletons = 8

Average dissimilarity = 0.66208726

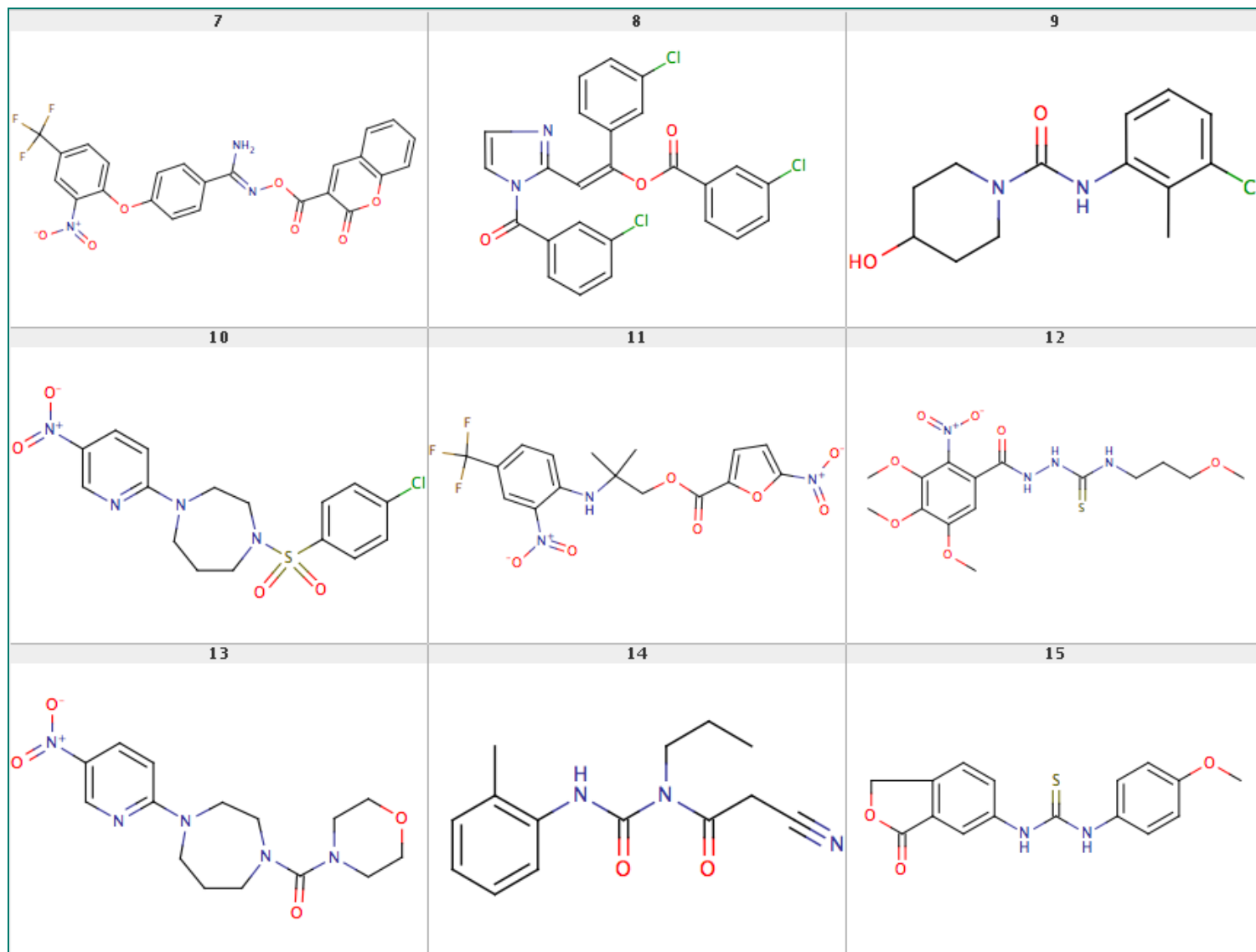
Minimum dissimilarity = 0.0

Maximum dissimilarity = 0.9411765

Parameter tuning

t	c	clusters	singletons
0.6	0.1	2	8
0.3	0.1	179	248
0.5	0.1	7	36
0.5	0.5	10	37
0.5	0.8	81	115

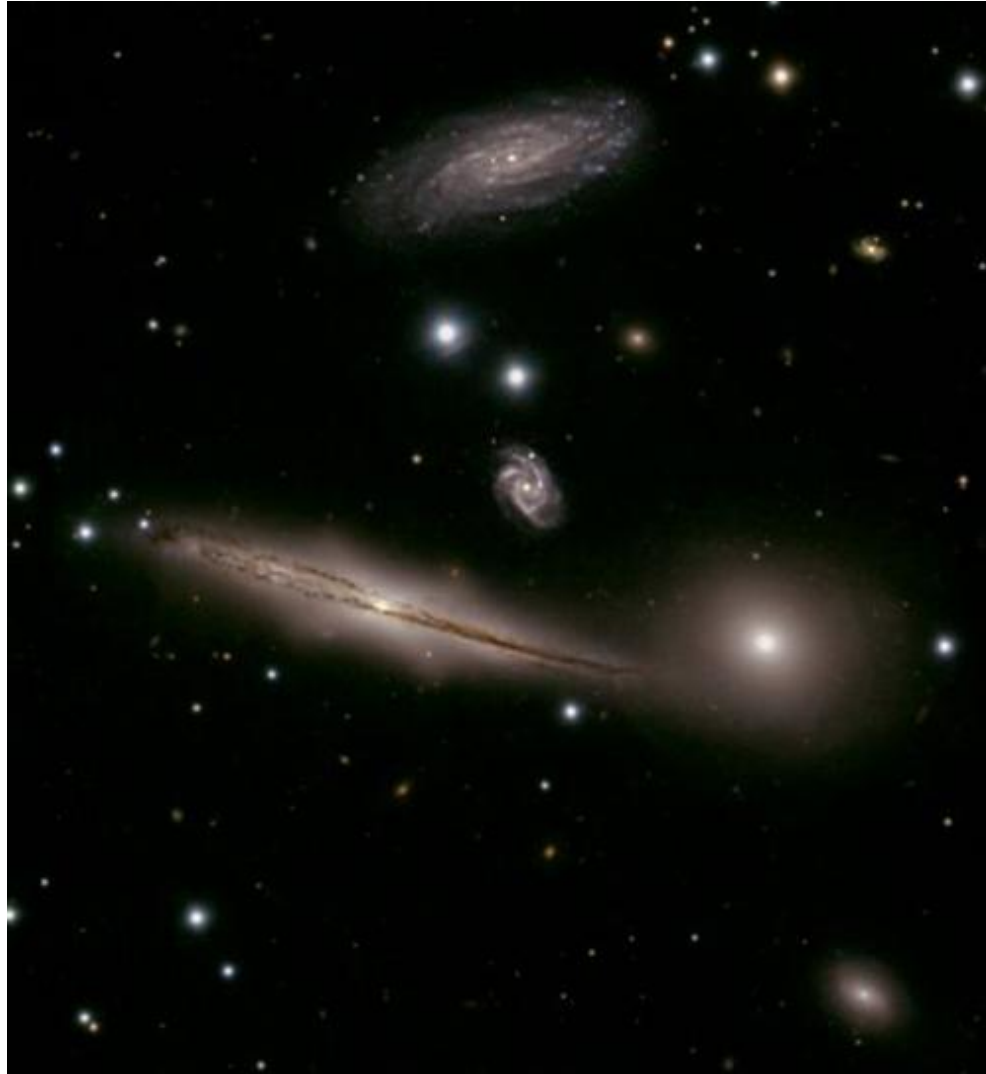
The most populated cluster



What's the matter with that?

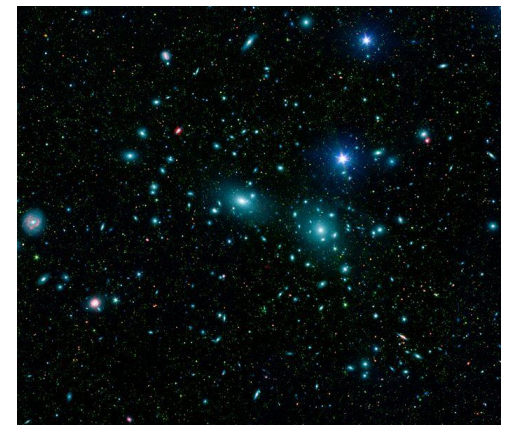
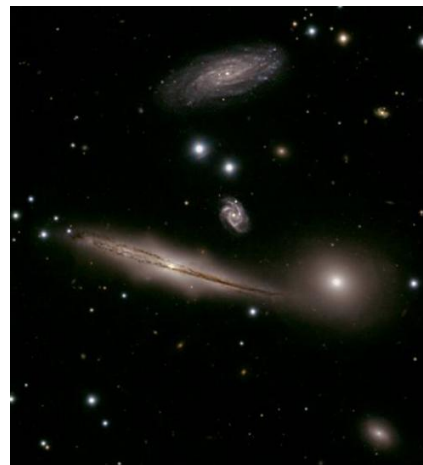
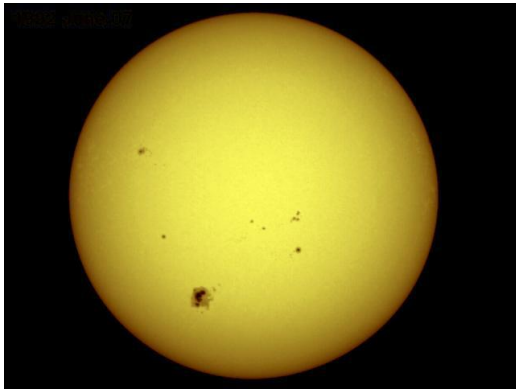
- tedious
 - parameter tuning in a trial-and-error fashion
- lack of interpretability
 - the algorithm does not provide explanation
- often do not meet chemists' expectation

Astronomers' job is easier



Why is clustering stars easy?

- stars have visible spatial arrangement
- distance between stars defines clusters, galaxies, ...



Images from Wikipedia, the free encyclopedia

Why is clustering molecules hard?

lack of innate spatial arrangement

- artificial arrangement
 - infinite types of chemical spaces
 - various ‘distance metrics’
 - usually high dimensionality (hard to visualize)
- various approaches, no superior one
 - “best method” depends on application area, and on actual data

What do we need?

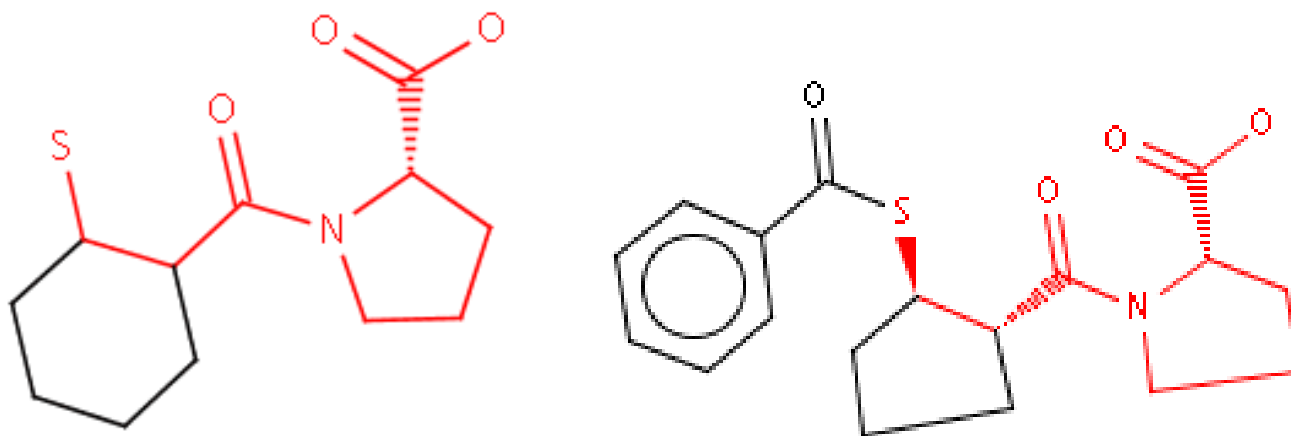
- no/few tuning
- easy to understand simple “explanation”

novel approach

- structure based clustering
- Maximum Common Substructure

Maximum Common Substructure

largest substructure shared by two molecules



Simple concept! More “human”, visual.

Yet hard (= expensive (= slow)) to compute.

Sub-structure searching

- *query* structure is known, it “only” have to be found as part of the *target* structure (subgraph isomorphism)
- graph isomorphism is even “simpler” yet NP-hard
 - finding the answer can take long (scales exponentially with respect to the number graph vertexes) in the worst case
 - validating an answer is fast

MCS

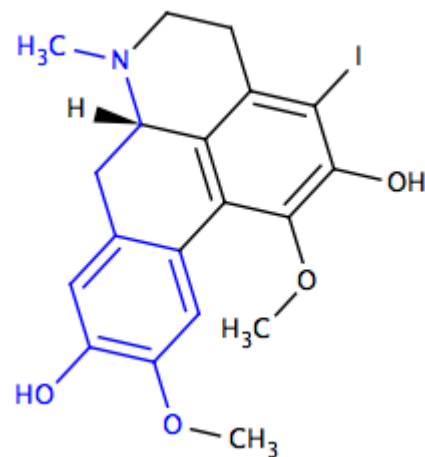
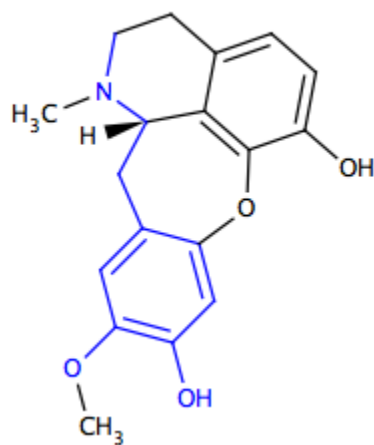
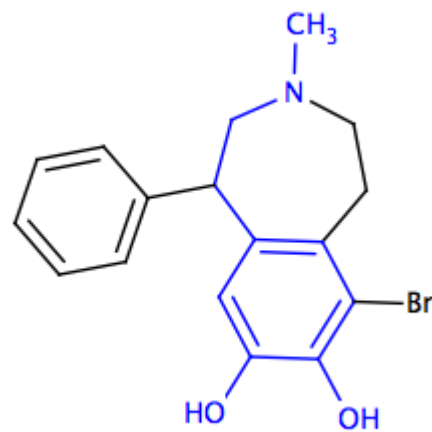
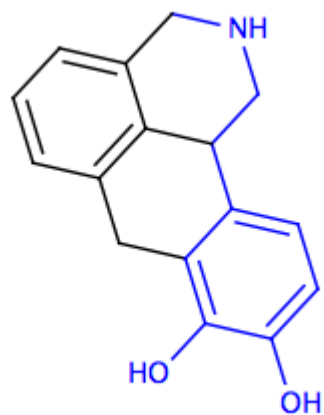
- “query” structure is not known
- all possible substructures need to be checked
 - even the number of substructures is exponential!

MCS algorithms

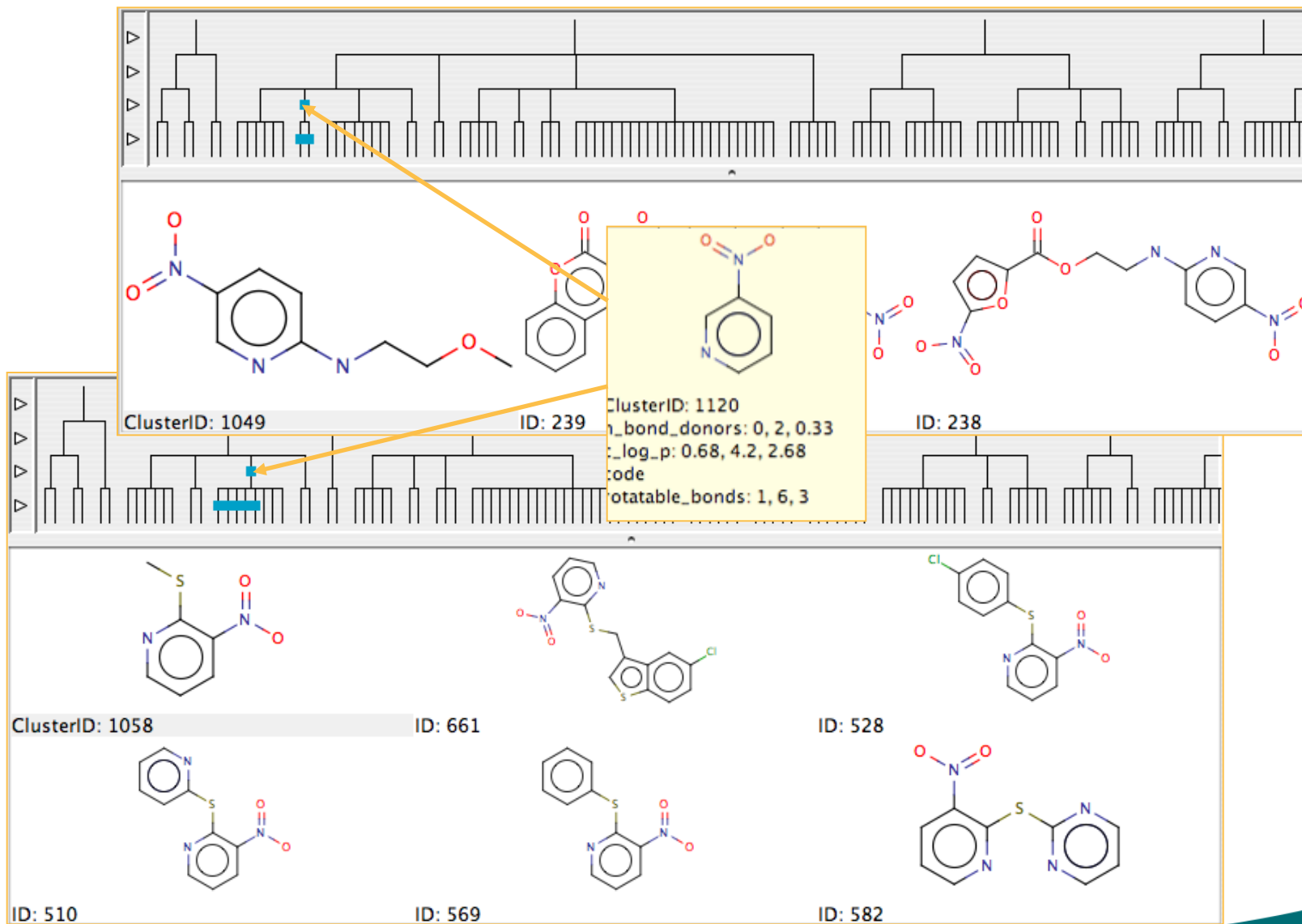
two camps

backtracking	clique detection
ad hoc	high mathematical elegance
average complexity is better than worst case	average complexity is same as worst case
dynamic heuristics	static (initial) heuristics
coloring is easy	coloring is hard
fuzzy matching	fussy matching

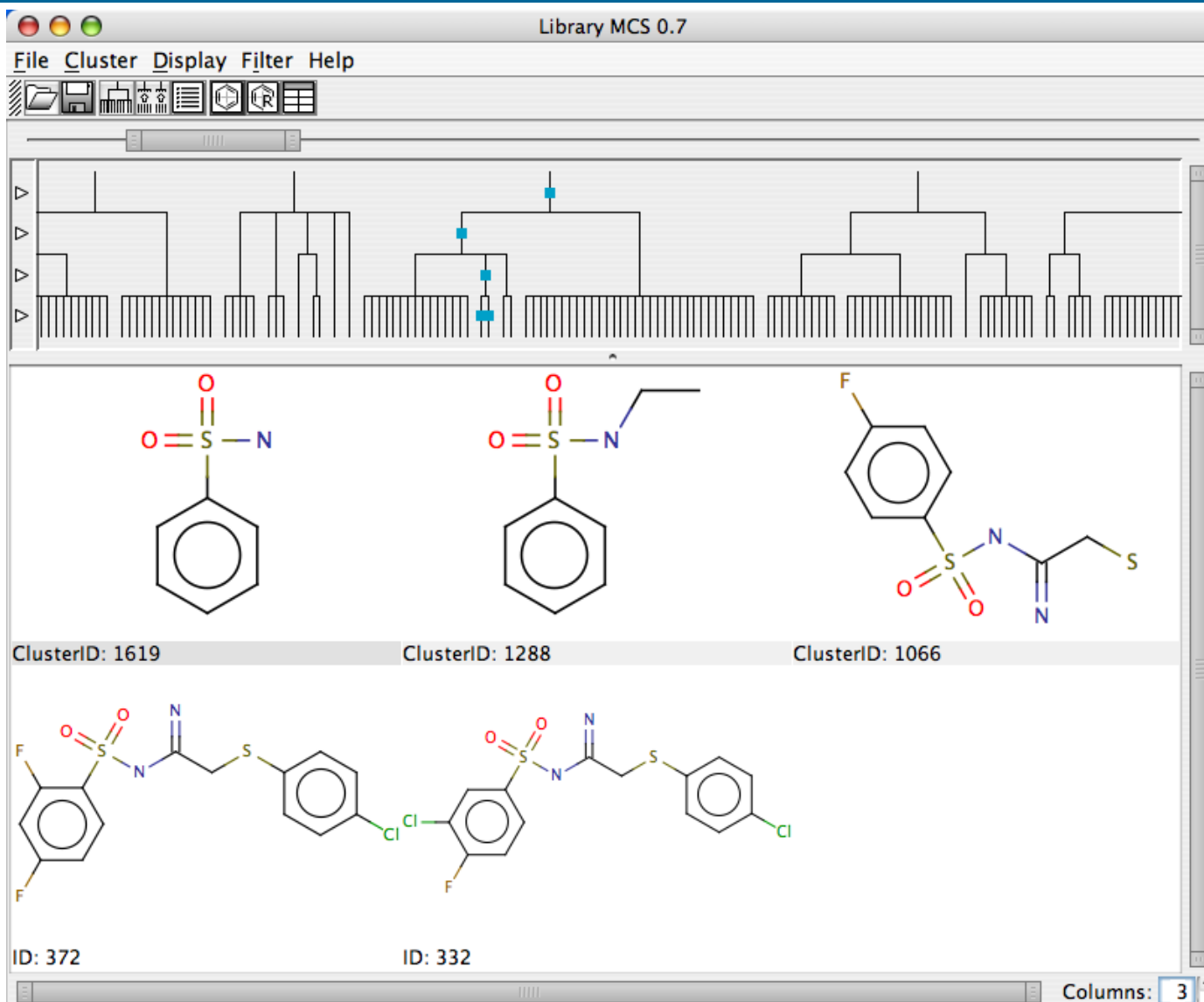
MCS of a structure set



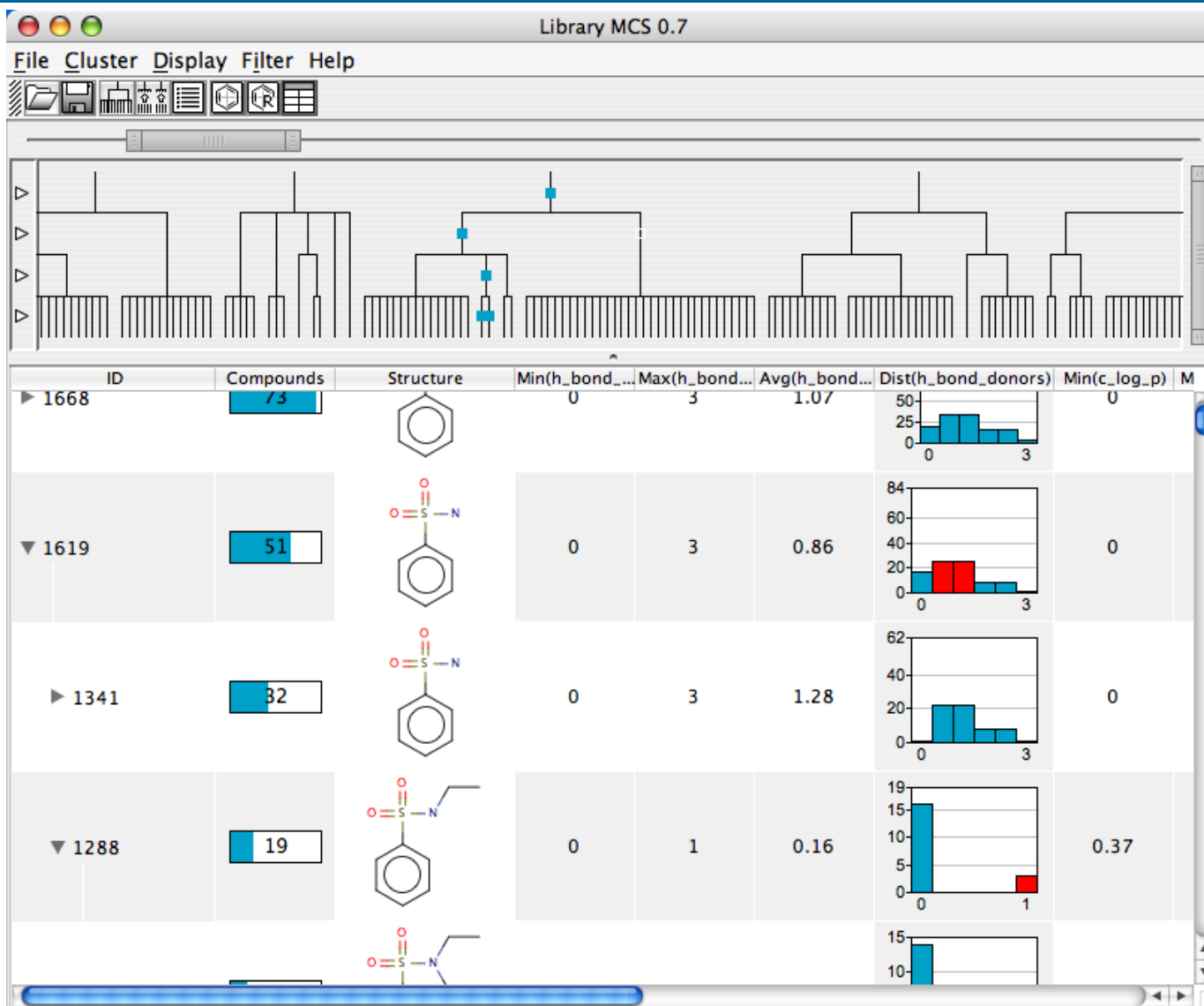
LibraryMCS: Hierarchical MCS



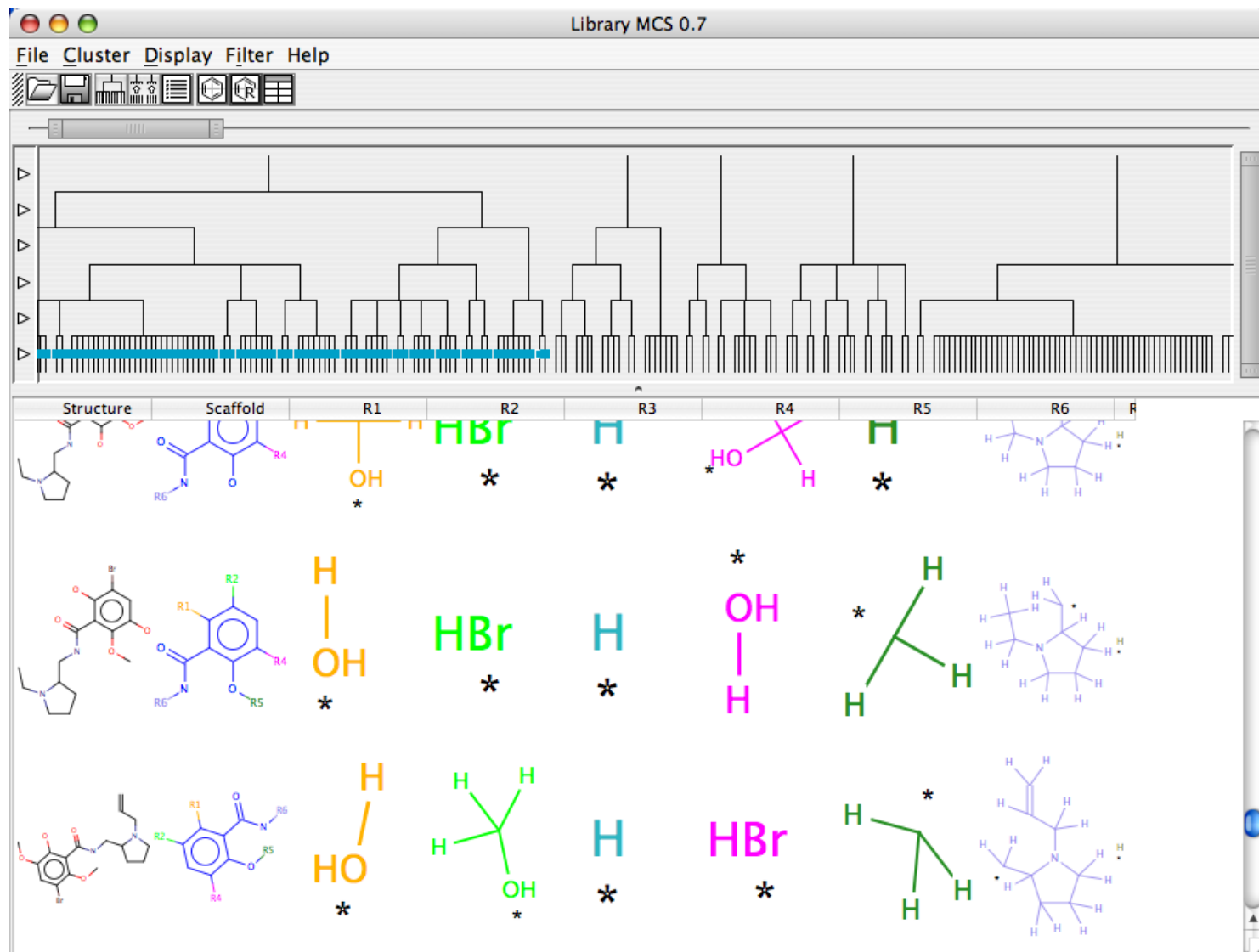
Intuitive visualization



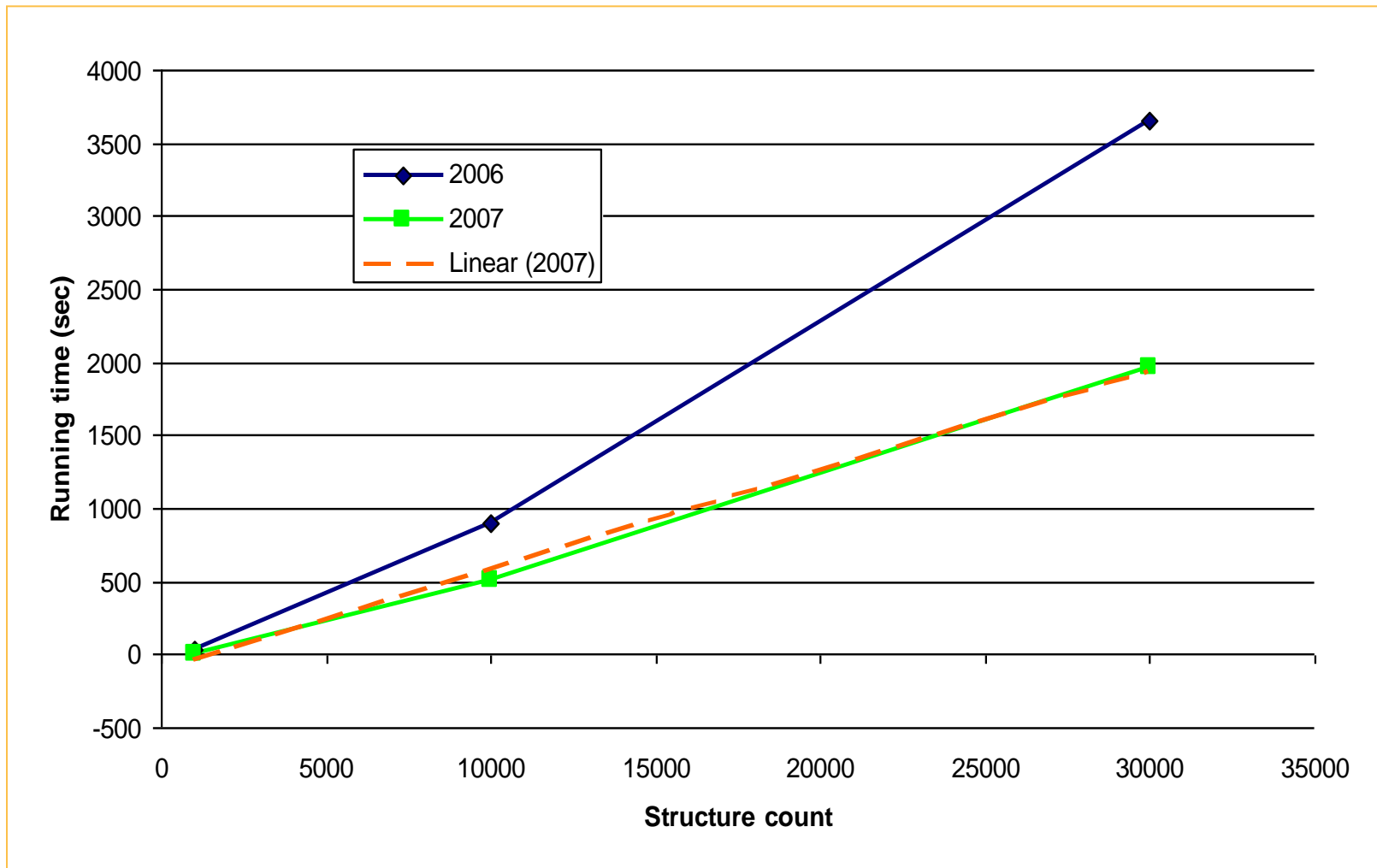
SAR table view



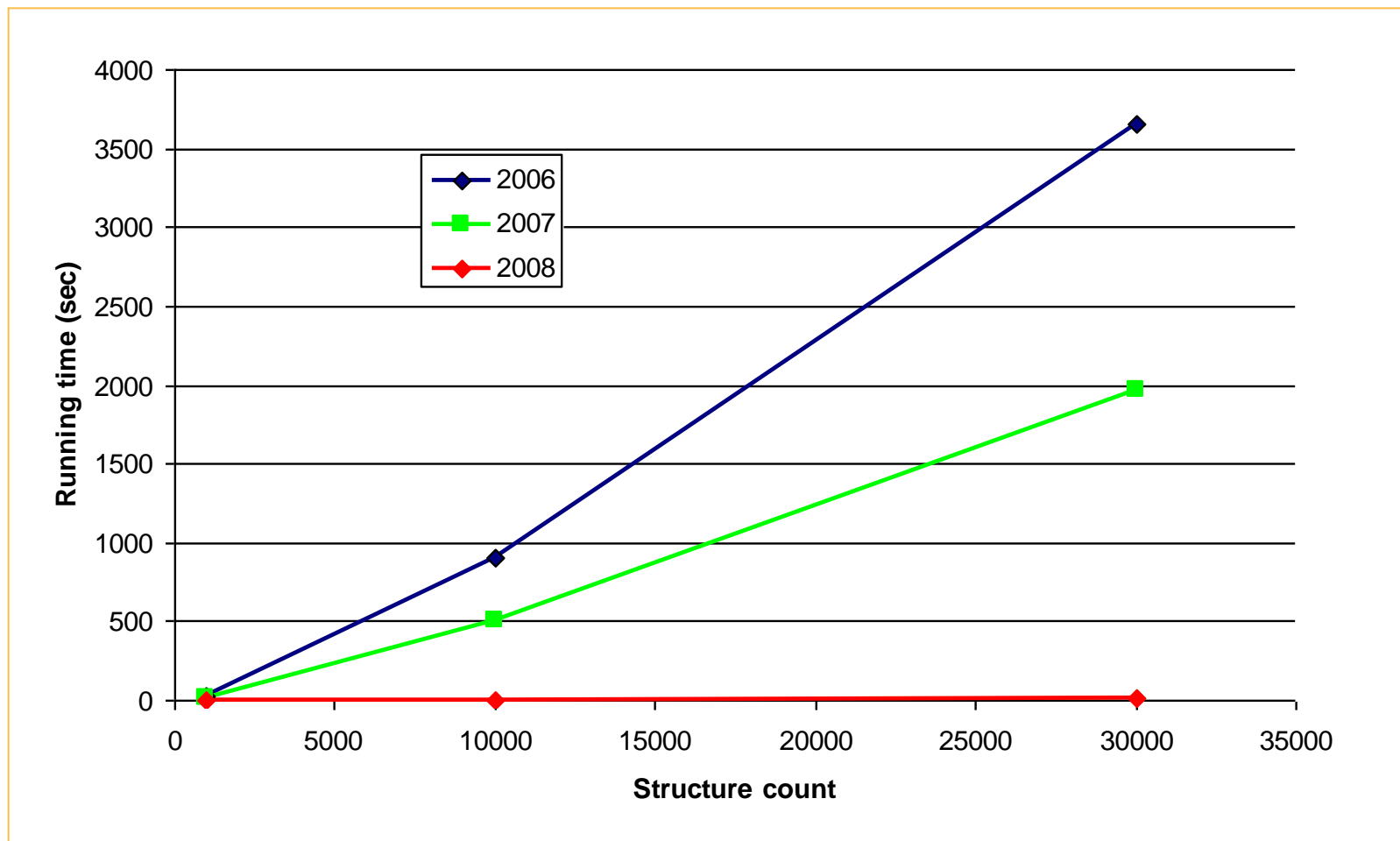
R-group decomposition



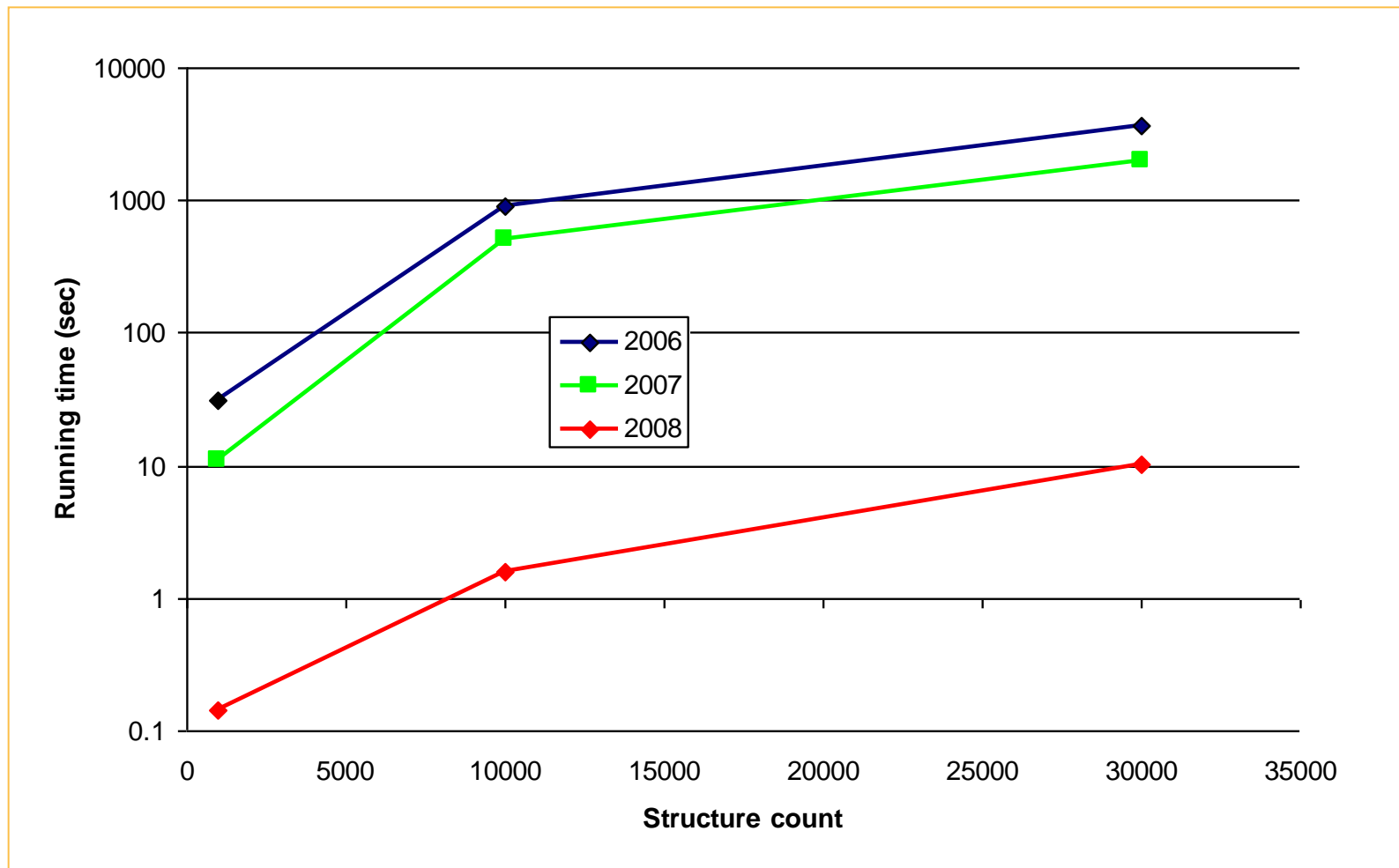
LibraryMCS scales linearly



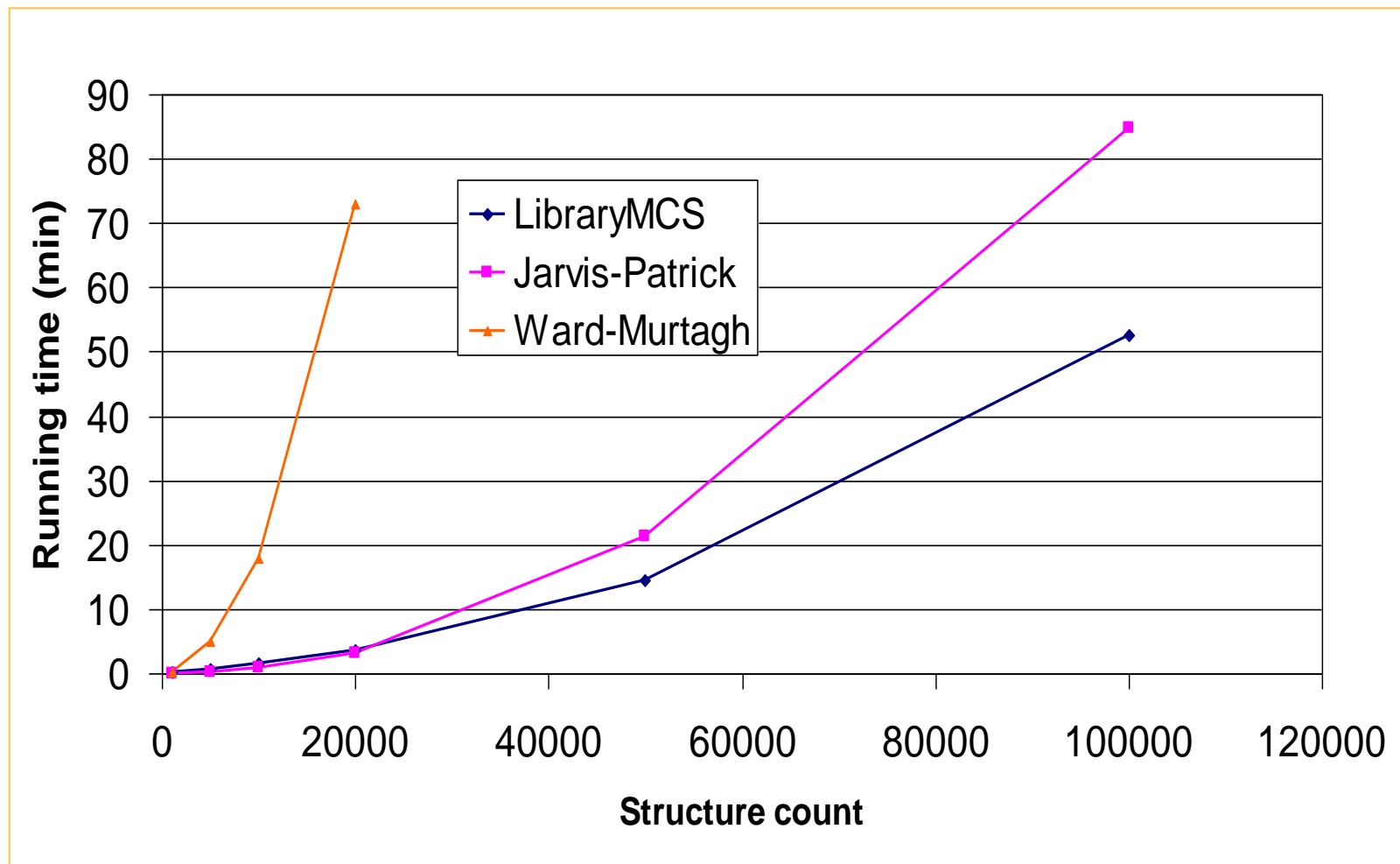
Speed-up achieved this year



Speed-up this year



Clustering performance comparison



Live demonstration

Clustering various types of structure sets

- VHTS hit sets
- focused libraries
- combi-chem libraries
- diverse sets
- corporate libraries

Behind performance

MCS search

- exhaustive
- heuristics
 - exact
 - inexact

Predictive MCS coupling in clustering

- all pairs are not feasible
- rich fingerprinting

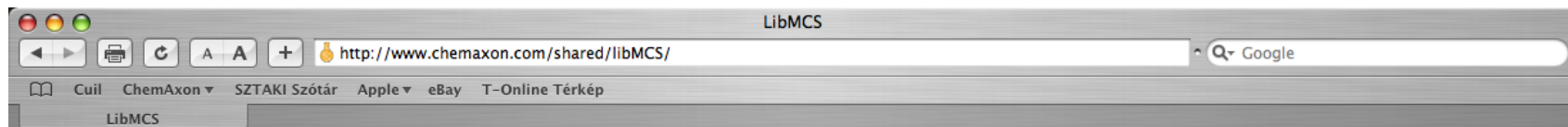
Live demonstration

Affect of use of heuristics

- on average < 10% misclassifications
- useful for obtaining birds-eye-view of a larger/diverse sets

Try it on-line

www.chemaxon.com/shared/libMCS



Library MCS

Version: 0.7

LibraryMCS clusters a set of chemical structures on a structural basis. Structures that share a common substructure are clustered together. The common substructure is identified by the clustering program, and it is always the largest one among all substructures found in the structure set. Such substructure is called the Maximum Common Substructure (MCS). No predefined fragments are applied in finding the MCS.

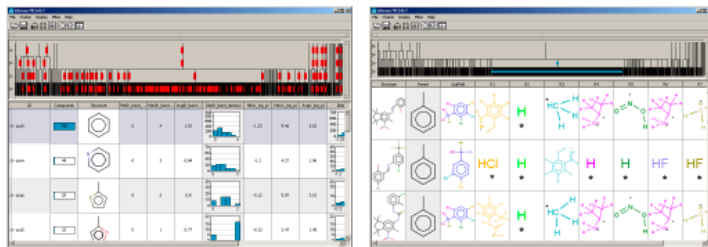
The clustering technique applied in LibraryMCS is hierarchical, that is, clusters of input structures are grouped into second level clusters, then these second level clusters are grouped again and so on, until a termination condition is reached (e.g. there is only one cluster left). These higher level clusters are also formed on a structural bases by recognizing the MCS of the constituting clusters. Numeric properties associated with the input structures are propagated through the clusters by calculating the minimum, maximum and average values of the properties, respectively. Beside of these a histogram per each property indicates how property values distribute in each cluster.

LibraryMCS is still under development, yet it is made available for trial and for early adaptors via this page. For the sake of easy deployment Library MCS is available both as an applet and as a Java Web Start application, thus there is no need to download and install the software. (Though it is also possible by [downloading JChem.](#))

Another important aim of this early product demonstration page is to allow future users shape the software according to their needs. All feature requests and more general suggestions are warmly welcome.

Library MCS with Java Web Start

Start LibMCS with Java Web Start.

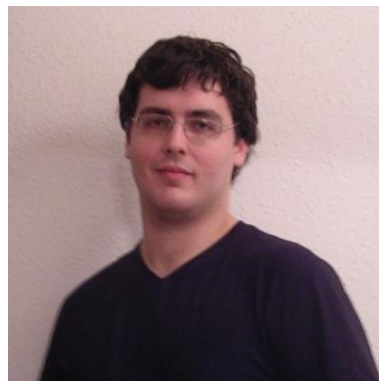


Start LibMCS with your input structures: <http://www.chemaxon.com/shared/libMCS/default.sdf>

Start!

Acknowledgements

Péter Vadász



Judit Vaskó-Szedlár

Gábor Imre

