DOI: 10.1002/cmdc.201700482

Comparison of Maximum Common Subgraph Isomorphism Algorithms for the Alignment of 2D Chemical Structures

Edmund Duesbury,*[a, b] John Holliday,[b] and Peter Willett[b]

The identification of the largest substructure in common when two (or more) molecules are overlaid is important for several applications in chemoinformatics, and can be implemented using a maximum common subgraph (MCS) algorithm. Many such algorithms have been reported, and it is important to know which are likely to be the useful in operation. A detailed comparison was hence conducted of the efficiency (in terms of CPU time) and the effectiveness (in terms of the size of the MCS identified) of eleven MCS algorithms, some of which were exact and some of which were approximate in character. The algorithms were used to identify both connected and disconnected MCSs on a range of pairs of molecules. The fastest exact algorithms for the connected and disconnected problems were found to be the fMCS and MaxCliqueSeq algorithms, respectively, while the ChemAxon_MCS algorithm was the fastest approximate algorithm for both types of problem.

Introduction

The maximum common subgraph (MCS) plays an important role in drug-discovery projects because it provides a simple, intuitive and chemically meaningful way of showing molecular similarity relationships by highlighting the substructural features common to two (or more) chemical graphs.[1–3] There are two distinct types of MCS. The connected MCS (hereafter cMCS) represents the largest single fragment (in terms of either the number of atoms or the number of bonds) common to two compounds when they are aligned; whereas the disconnected MCS (hereafter dMCS) can contain multiple fragments and is the set of fragments that maximises the number of atoms or bonds in the MCS. The connected and disconnected forms of the MCS are illustrated in Figure 1. There are two additional subdivisions of the MCS: the maximum common edge-induced subgraph (MCES) representing all the edges (i.e., bonds) between two graphs; and the maximum common node-induced subgraph (MCIS) representing the nodes (i.e., atoms) in common between two graphs with their respective end-points preserved. In this work, we have focused on the more intuitive MCES, with example applications including reaction mapping,[4] clustering,[5] matched molecular pairs analysis,[6,7] chemical space network representation,[8] and three-dimensional molecular alignment[9] inter alia.

Determination of the MCS is an example of a non-deterministic polynomial time complete (NP-complete) problem. If a polynomial solution can be found for one NP-complete problem, then all other NP-complete problems can be transformed to become solvable in polynomial time too.[10,11] As an example of the MCS being an NP-complete problem, finding the MCS can be transformed into finding the maximum clique in the modular product of two graphs.[12,13] Considerable progress has been made in improving the efficiency of clique algorithms but no polynomial-time solution currently exists that would enable the detection of an exact MCS solution. There has, however, been recent work on the development of fast and approximate solutions to finding the MCS. Building on earlier work,[14–16] one approach involves the use of the topologically constrained dMCS (hereafter tdMCS)[17,18] Finding the tdMCS relies on deleting those edges in the modular product of two

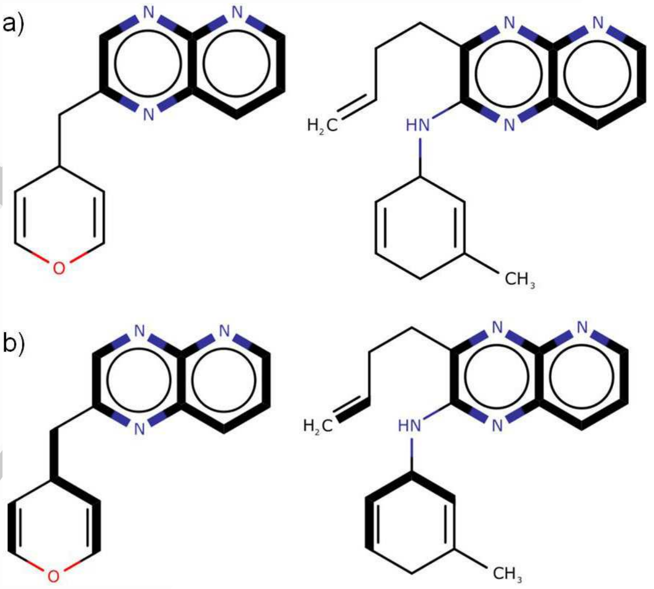


Figure 1. Types of MCES (the MCS in each pair of compounds represented by bold edges): a) cMCS between the two molecular graphs; b) dMCS between the same two molecular graphs.

[a] Dr. E. Duesbury
Computer-Aided Drug Design, UCB Pharma, 208 Bath Road, Slough, SL1 3WE (UK)
E-mail: Edmund.duesbury@ucb.com

[b] Dr. E. Duesbury, Dr. J. Holliday, Prof. Dr. P. Willett
Information School, University of Sheffield, 211 Portobello, Sheffield, S1 4DP (UK)

Supporting information and the ORCID identification number(s) for the author(s) of this article can be found under: https://doi.org/10.1002/cmdc.201700482.

chemical graphs that correspond to pairs of node that have differing topological distance differences to each other. The tdMCS has been found to yield more realistic alignments than the standard dMCS, despite it being significantly faster to detect, and it has also been suggested that it is more effective than the dMCS in virtual screening applications:[15,17] As in the work of Kawabata,[17] we shall use Θ to describe the allowed topological distance difference, as illustrated in Figure 2. A topological distance difference constraint (corresponding to $\Theta = 0$) was also one of the comprehensive, but complex, set of heuristics introduced by Raymond et al.[19] to expedite the identification of an MCS equivalent to, or at least similar in size to, the dMCS. These heuristics are applicable to several of the algorithms considered here and are described in detail by Duesbury in his doctoral thesis,[20] who found that use of the heuristics did indeed yield reductions (and substantial reductions in some cases) of the running times of these algorithms. However, this increase in efficiency was often accompanied by a reduction in the size of the MCS that was identified, and the results for this approach (which is referred to as hMCS by Duesbury) have hence not been included in the results reported here.

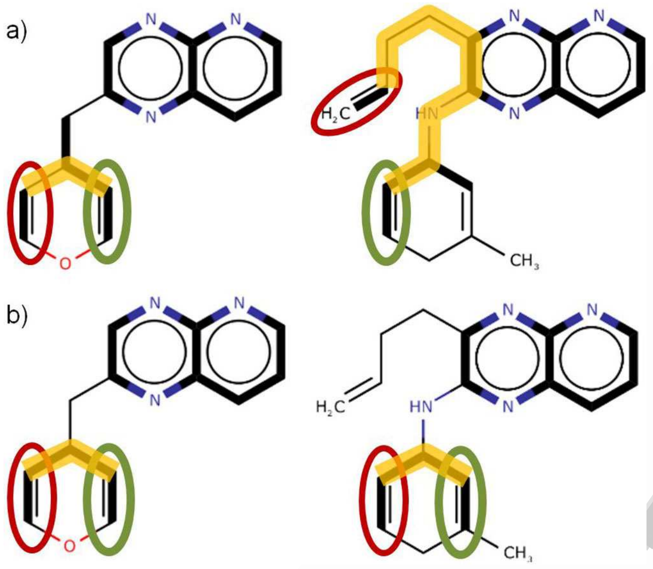


Figure 2. a) The dMCS between two chemical graphs, and b) the corresponding tdMCS ($\Theta = 0$). In this example MCS, some of the bonds in the two compounds have been mapped to yield an alignment that in geometric terms would be chemically unfeasible. The difference in distance between the two circled bond pairs in the dMCS is 5 (7−2). By limiting the size of this difference we can decrease the search space and yield more sensible alignments, as represented by the tdMCS.

There have been relatively few empirical comparisons of MCS algorithms in chemoinformatics and those that have been reported are either now quite dated, e.g., the studies by Brint and Willett[21] and by Gardiner et al.,[22] or have involved only a limited number of different algorithms, e.g., the studies by Rahman et al.[23] and by Hariharan et al.[24] (who consider the alignment of multiple molecules rather than just two as here). This article reports an extended comparison of eleven different MCS algorithms, both in terms of time performance (i.e., the efficiency of the algorithms), and MCS size (i.e., their effectiveness) when used for the alignment of pairs of 2D chemical structures using the three different definitions of an MCS (i.e., cMCS, dMCS and tdMCS) described above.

The algorithms tested in this study are listed in Table 1, where the names are those in the original publication. They have been characterised in the table by being either exact or approximate in nature, and by the number and types of MCS that they are able to identify. Details of the algorithms can be found in the original cited articles[17–19,23,25–31] and in the thesis by Duesbury.[20] The performance of these algorithms has been assessed using the two datasets that are shown in Tables 2 and 3 and that are described in detail in the Experimental section of the paper.

Table 1. Types of MCS algorithm included in this study, with the types of MCS detection for which they are applicable denoted by an X. The kcombu algorithm identifies *k* (a user-defined value) MCSs.

Output	Algorithm and reference	cMCS	dMCS	tdMCS	Number of MCSs
Exact	VFLibMCS[23]	X			>1
	Small Molecule	X			>1
	Subgraph Detector[23]				
	fMCS[25]	X			1
	MaxCliqueSeq[26]		X	X	1
	Bron–Kerbosch[27]	X	X	X	all
	Carraghan–Pardalos[28]		X	X	1
	RASCAL[19,29]		X	X	1
Approximate	CDKMCS[23]	X			>1
	kcombu[17]	X	X	X	*k*
	consR[30]		X		1
	ChemAxon_MCS[18,31]	X	X		1

Results and Discussion

The results are shown in Figures 3 and 4, in addition to Tables SI1–SI8 (Supporting Information). In the aforementioned Figures, violin plots have been used to represent time and MCS size distributions for each algorithm-MCS type combination. A violin plot is an extension of the box-and-whisker plot, where each "violin" is a plot of the density function of a histogram produced from the data.[44] This has some advantages over simple box plots, for it shows how the data are distributed (for example, a box plot would not distinguish a bimodal distribution from a uniform one). In this work, the maximum width for each "violin" is constant, the width representing the probability density at a given point. Maxima and minima of the compound times are represented by the corresponding extremes for each "violin." The black dots represent the median of the data. The tables show the same information as the violin plots, but in more detail, should the reader desire. The efficiency is denoted by the median run time (in seconds) and the median absolute deviation averaged over the ten runs for each algorithm (the latter is absent if the algorithm failed to complete

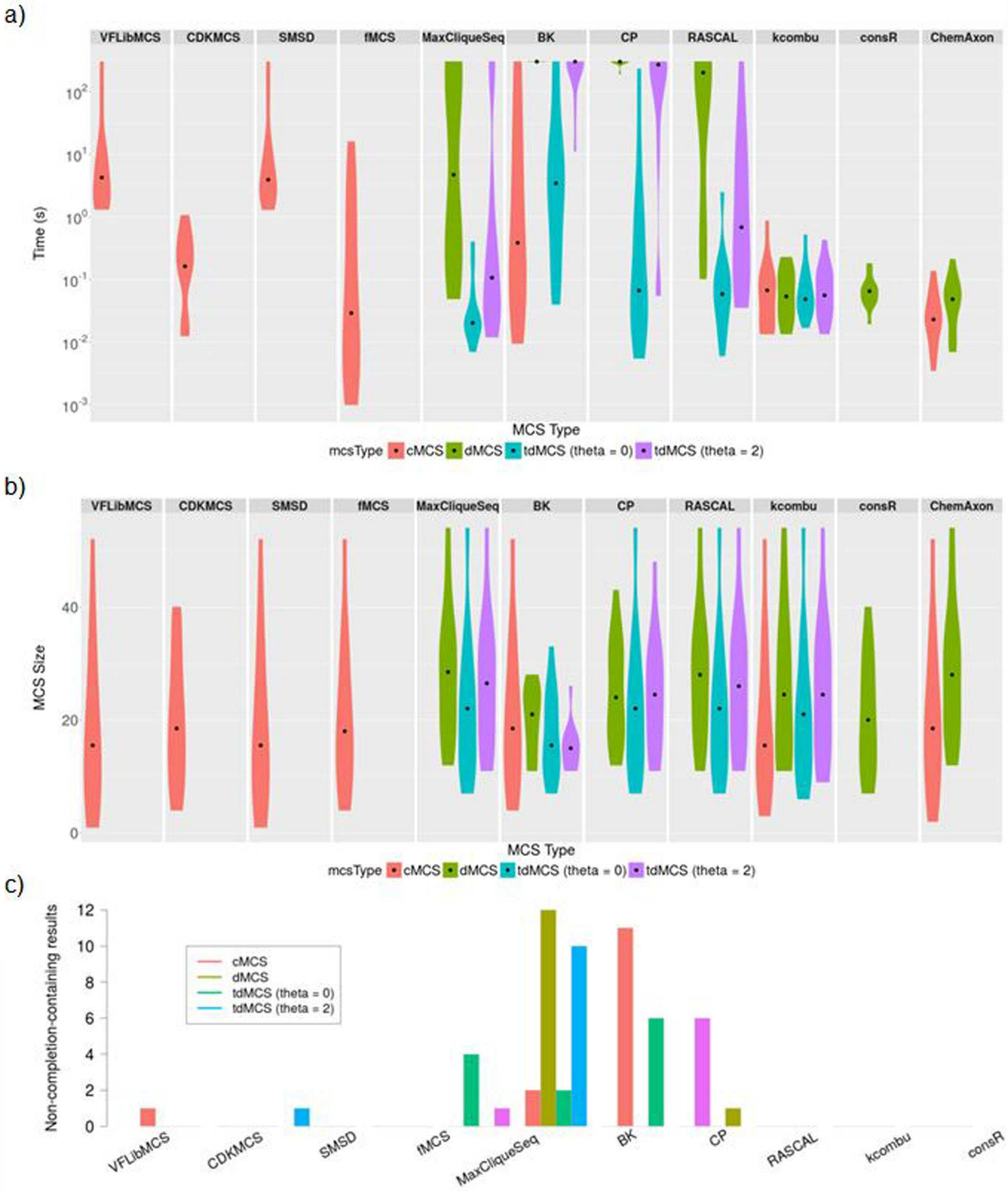


Figure 3. Charts showing performance information of each algorithm (including the same algorithm ran on different MCS types) for the S, N, and M compound pairs. a) Time taken; b) MCS size (in bonds); c) number of non-completions for a method.

all of the runs), and the effectiveness by the size (in terms of numbers of bonds) of the MCS that was identified. The results for the most efficient and the most effective performers are bold-faced. If a time has been italicised then the median was calculated excluding any results that failed to run to completion. Results that only comprised non-completions are also italicised, with the median absolute deviation being set to zero as it was not calculated.

We consider first, in Table SI1 and Figure 3, the results for the pairs of molecules in Table 2 when searching for the dMCS. It will be seen that BK was slowest in all of the cases tested here, an unsurprising result in that it is designed to retrieve all maximal cliques rather than just a single maximum clique. Carraghan–Pardalos (hereafter CP) was only marginally superior (obtaining a solution within the time-limit for one of the ten pairs of molecules) but normally found larger solutions than Bron–Kerbosch (hereafter BK). The other two exact algorithms—RASCAL and MaxCliqueSeq—obtained solutions for most of the pairs, with the latter normally being the faster of the two. Turning now to the three approximate dMCS algorithms, ChemAxon_MCS (hereafter CA_MCS) was notably faster than the other two (kcombu and consR) and (hardly surprisingly) than the exact algorithms: it identified the largest MCS for ten of the pairs and had the fastest median run time for seven of them, thus demonstrating the effectiveness of the heuristics in this algorithm. Of the other two approximate algorithms, kcombu normally out-performed consR. Overall, MaxCliqueSeq and CA_MCS would appear to be the algorithms of choice for this type of MCS search.

The tdMCS results for $\Theta = 2$ and $\Theta = 0$ are shown in Tables SI2 and SI3, respectively, for BK, CP, MaxCliqueSeq, RASCAL, and kcombu; it was not possible to make the necessary modifications to the basic dMCS algorithms for CA_MCS and consR. The increased constraints mean that the run times here are faster and the maximum sizes smaller than in Table SI1. This is particularly noticeable for BK (which was now able to provide solutions for two of the pairs when $\Theta = 2$ and for ten of them when $\Theta = 0$) and to a lesser extent for CP. The approximate kcombu was generally the fastest for $\Theta = 2$ (albeit often with a markedly sub-optimal MCS) but this was not the case for $\Theta = 0$. Overall, MaxCliqueSeq gave the best performance here: it always found the largest MCS and was, at least for $\Theta = 0$, also often the fastest algorithm.

The seven sets of cMCS results are shown in Table SI4. CDKMCS was faster than the other two CDK algorithms (VFLibMCS and Small Molecule Subgraph Detector (hereafter SMSD)) but the best performers here were clearly fMCS and CA_MCS. The approximate nature of the latter meant that the MCSs that it detected were occasionally smaller than those identified by fMCS (and also those identified by BK but the run-times here were totally uncompetitive); against that fMCS was much slower than CA_MCS, for two of the symmetric molecule pairs (S4 and S5). Even so, fMCS is probably the algorithm of choice for this particular application.

Figure 4 and Tables SI5–SI8 give the analogous performance information for the 25 pairs of molecules in Table 3. The runs were generally faster and had less non-completions than for the pairs of molecules in Table 2. MaxCliqueSeq was again consistently superior to the other exact clique detection algorithms in terms of both MCS size and speed; it also outperformed the approximate CA_MCS in terms of size, though was still slower in the dMCS runs. The CP results here were notably better than those discussed previously, nearing the other two high-performance algorithms in size and time performance (although it had more non-completions). In particular, it was faster than all of the other applicable algorithms for tdMCS ($\theta = 0$), suggesting that CP is most appropriate for simpler pairs of molecules and MCS solutions.

Unfortunately, some inconsistencies in the results for the exact cMCS algorithms shed questions as to whether the algorithms actually provide exact answers. For example, the VFLibMCS and SMSD algorithms failed to identify the MCS for the N1 pair, and the same algorithms plus fMCS failed for M2, despite the algorithms converging. Whilst the source code for fMCS was translated from Python to Java and mistakes may have occurred during the translation, the original source code for VFLibMCS and SMSD was used. For the Franco compound pairs, 92a and 96a present particularly interesting outliers as the VFLibMCS and SMSD algorithms actually outperformed other exact solutions. Presumably, this can only reflect abnormalities in the coding of these algorithms. The same can be said for RASCAL of upon inspection of the three incomplete convergences for the Franco compounds for tdMCS ($\Theta = 0$), perhaps hinting at a missing detail in the original algorithm lacking in our program version.

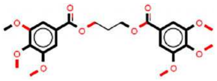
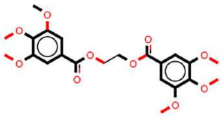
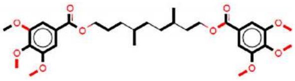
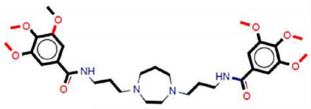
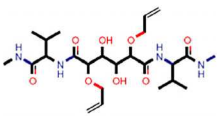
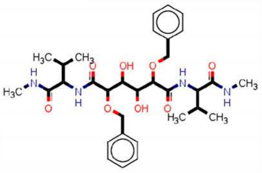
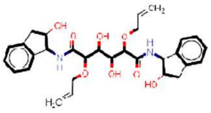
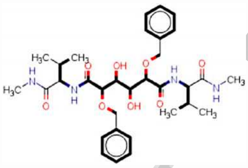
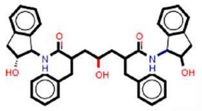
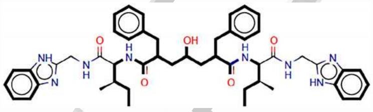
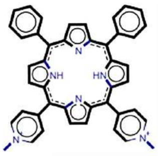
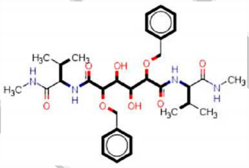
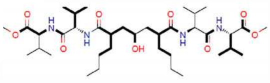
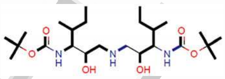
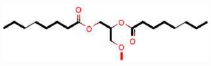
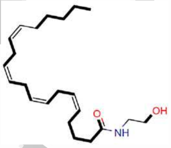
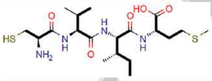
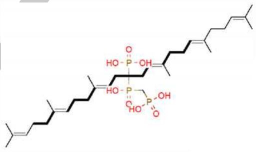
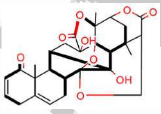
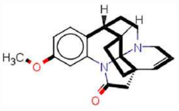
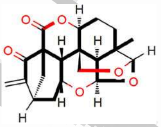
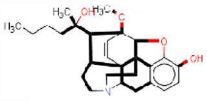
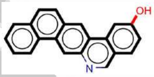
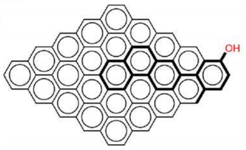
Conclusions

Benchmark studies in chemoinformatics, e.g., comparisons of fingerprint-types or similarity coefficients etc., often conclude with a recommendation for a "best buy" approach, and this work was undertaken with the expectation that it would be possible to make analogous recommendations here. However, the results in Figures 3 and 4, in addition to Tables SI1–SI8, demonstrate clearly that the algorithm of choice is determined in large part by the precise nature of the MCS that is to be identified. Specifically, we suggest that the MaxCliqueSeq algorithm described by Depolli et al.[26] be used for aligning pairs of molecules by means of the dMCS and the fMCS algorithm described by Dalke[25] be used for aligning pairs of molecules by means of the simpler cMCS definition. It is possible to increase substantially the speed of the first of these by the imposition of progressively stricter topological distance constraints, though the resulting tdMCS will often be smaller than the dMCS. Both MaxCliqueSeq and fMCS are exact algorithms that are guaranteed to identify the true MCS (given sufficient time in some cases): of the approximate algorithms that were tested, CA_MCS provides an efficient and an effective way of identifying both the cMCS and the dMCS.

Experimental Section

The hardware used in this study featured an Intel(R) Core^{TM} i7-2600 CPU @ 3.40 GHz processor with 16 GB of DDR3 RAM clocked at 1333 MHz, running Kubuntu 13.10. The Konstanz Information Miner (KNIME) 2.8.2[42] running Java 1.6 was used for all experimental aspects in this study, and the Chemistry Development Kit 1.5.3 (CDK) was used for all chemoinformatics functionality, unless other-

Table 2. Compounds selected for use in the first dataset for benchmarking.

Number	Compound 1	Compound 2	Reference(s)
S1			[29]
S2			[19]
S3			[35]
S4			[35]
S5			[36]
S6			[37]
N1			CHEMBL39130, CHEMBL46316
N2			CHEMBL26440, CHEMBL15848
N3			CHEMBL55423, CHEMBL1204752
M1			[38,39]
M2			[40,41]
M3			N/A

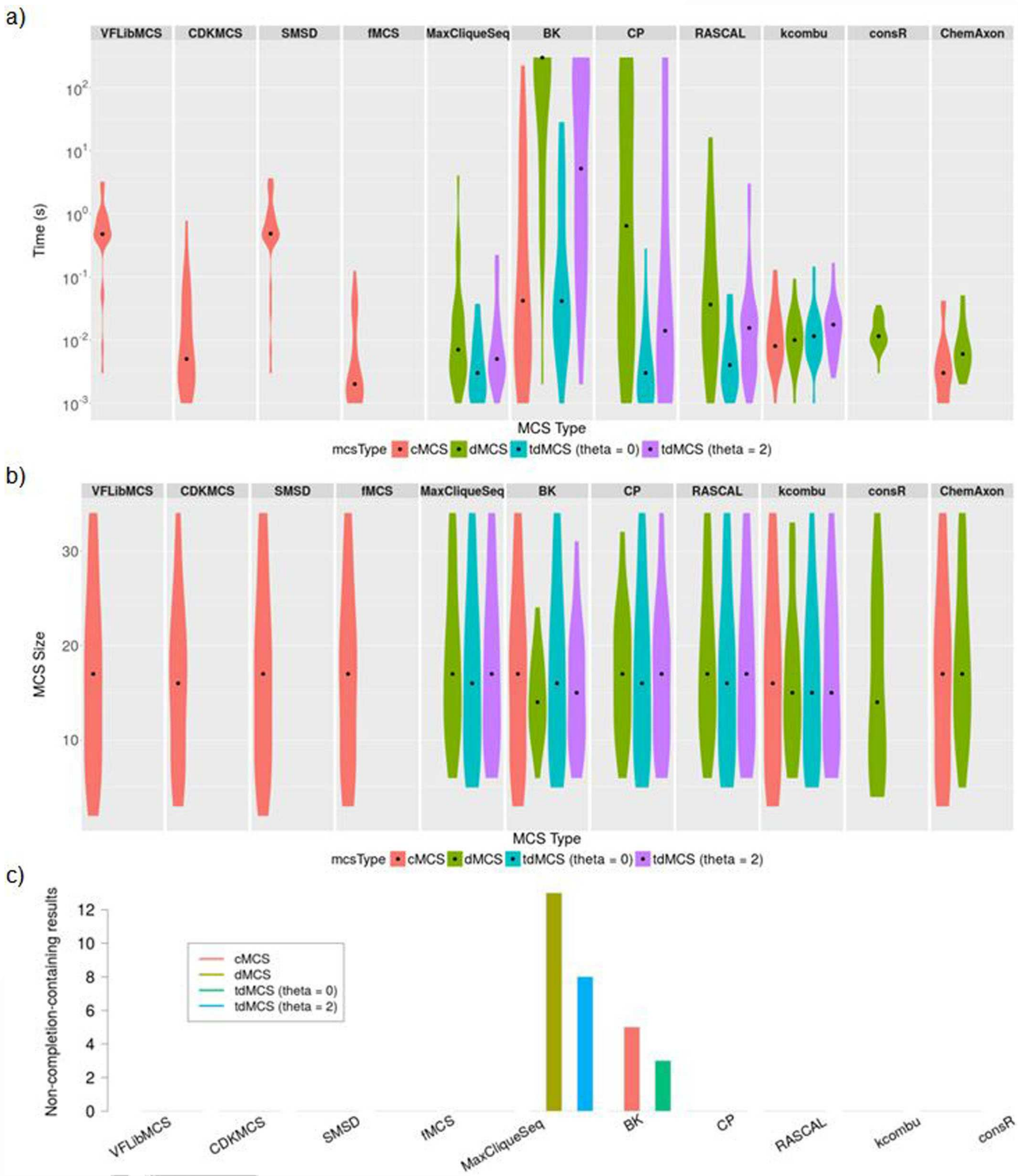


Figure 4. Charts showing performance information of each algorithm (including the same algorithm ran on different MCS types) for the Franco compound pairs. a) Time taken; b) MCS size (in bonds); c) number of non-completions for a method.

wise noted[43] All eight CPU cores were used for each algorithm. R with the ggplot2 package was used to produce plots.

The algorithms tested in this study are listed in Table 1. They were implemented as closely as possible to how they were described in the original publications, using Java for the KNIME platform (these implementations are detailed by Duesbury).[20] For algorithms

Table 3. Franco compound pairs to be evaluated, sorted in descending order of mECFP4 similarity, as depicted in the "Similarity" column.

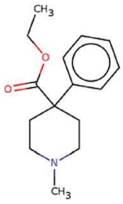
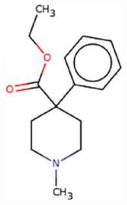
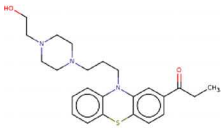
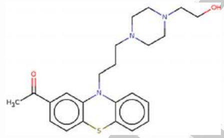
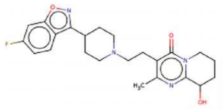
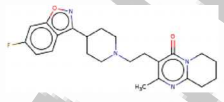
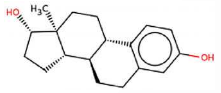
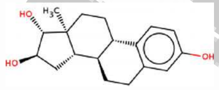
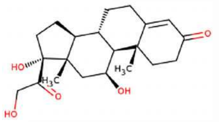
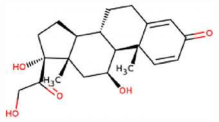
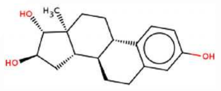
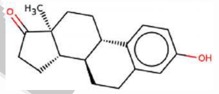
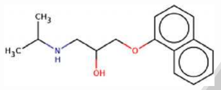
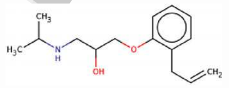
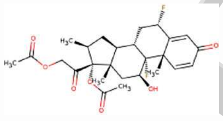
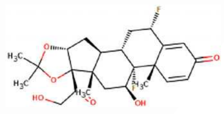
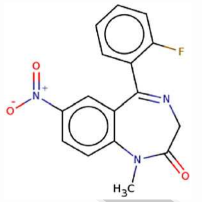
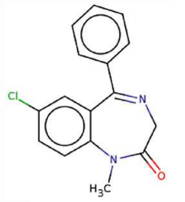
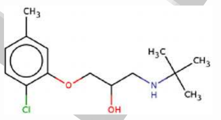
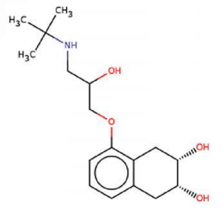
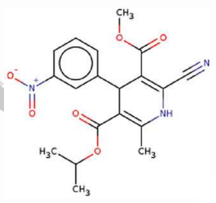
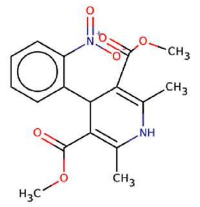
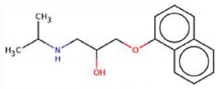
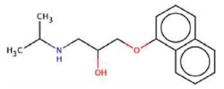
Number[a]	Compound 1	Compound 2	Similarity
F80a			1.000
F89a			0.824
F2a			0.739
F57a			0.723
F92a			0.678
F31a			0.636
F19a			0.587
F100a			0.544
F91a			0.500
F94a			0.481
F72a			0.444
F18a			0.415

Table 3. (Continued)

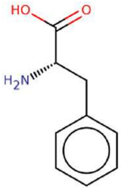
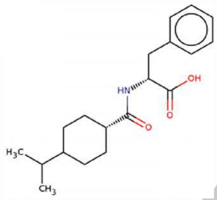
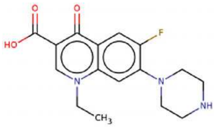
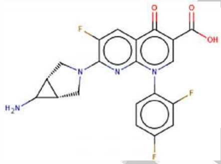
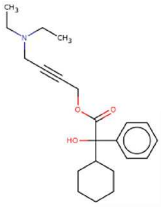
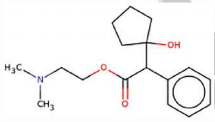
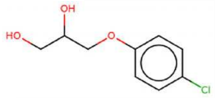
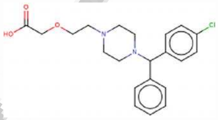
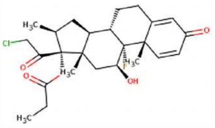
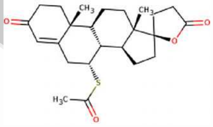
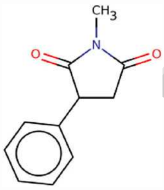
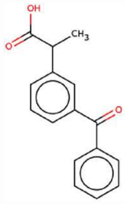
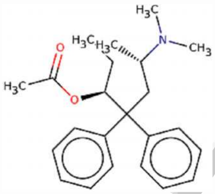
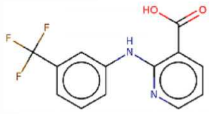
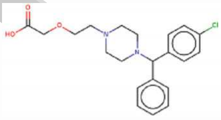
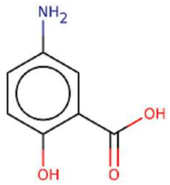
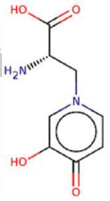
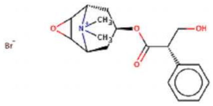
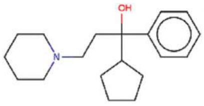
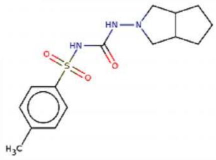
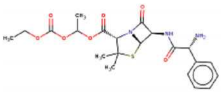
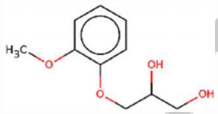
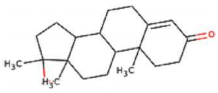
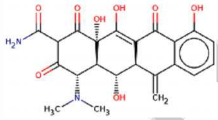
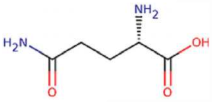
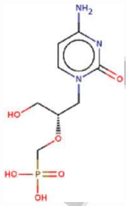
Number[a]	Compound 1	Compound 2	Similarity
F34a			0.391
F47a			0.366
F69a			0.279
F28a			0.214
F96a			0.186
F50a			0.170
F8a			0.152
F13a			0.138
F45a			0.134

Table 3. (Continued)

Number[a]	Compound 1	Compound 2	Similarity
F12a			0.129
F30a			0.117
F20a			0.105
F88a			0.091

[a] Name of the compound pair that the authors originally assigned.

where the source code existed, but in a different language, the source code was translated manually to Java (including BK, CP, fMCS, MaxCliqueSeq, and kcombu). The sole exception was ChemAxon_MCS (hereafter CA_MCS), which is a proprietary algorithm for which we do not have precise details, but which is based on the procedures described by Englert and Kovacs[18] and by Grosso et al.[31] It is important to note that the transcription and/or translation to Java code for each algorithm, was done to eliminate potential performance differences arising from using different programming languages. However, this does not perfectly reflect the original algorithm's performance—for instance, MaxCliqueSeq was originally implemented in C++, thus the original algorithm generally is expected to perform faster than the Java version implemented here.

All of the clique detection algorithms (BK, CP, MaxCliqueSeq and RASCAL), fMCS, SMSD and VFLibMCS were implemented with a time-out condition. This was not done for the four approximate algorithms, as these did not exceed the time constraint (and CA_MCS is proprietary). Of the exact clique-detection algorithms implemented here, Hariharan et al.[24] describe a cMCS solution for the Bron–Kerbosch algorithm; however, it was not possible to apply their modification to the other three such algorithms due to the particular heuristics that each of them use, and we were thus able to compute only the dMCS and tdMCS for these algorithms.

We used two very different datasets to assess the performance of the eleven MCS algorithms when they were used to align pairs of 2D molecules. The efficiency of each algorithm was assessed by means of the average run-time taken to identify the maximum common substructure, and the effectiveness by the size in terms of the number of bonds of that common substructure. Factors that are known to influence search time include the sizes of the graphs that are being compared, the level of symmetry in the graphs and the average degree of the nodes comprising the graphs.

The first dataset comprised twelve pairs of compounds, these being in the three subsets comprising Table 2 (where the boldfaced bonds constitute the dMCS in each case). The first subset (denoted by S1–S6) consists of pairs of highly symmetric compounds, as Raymond et al. noted that the degree of symmetry was negatively correlated with the speed of MCS detection.[19] There are then three pairs of non-cyclic molecules (**N1**–**N3**), where it will be seen that the dMCS is quite fragmented, and the table is completed by three pairs of miscellaneous molecules. There are two pairs of nonplanar molecules (**M1** and **M2**), as Dalke has suggested that planar and nonplanar graphs might yield different performance statistics,[32] and the final pair, **M3**, has been chosen to see how well the algorithms deal with large, highly connected graphs.

The pairs of molecules in Table 2 were chosen specifically to test the performance of the algorithms, and many of them are hence rather more complex than the sorts of molecules that might be encountered in conventional MCS applications such as reaction indexing or pharmacophore mapping. The second dataset hence contained pairs of drug-like molecules from the DrugBank 3.0 database.[33] Franco et al. selected 100 such pairs from the database as part of a study to assess the extent of the agreement between human-based and fingerprint-based judgements of molecular similarity.[34] These 100 pairs were sorted here into descending similarity order (based on RDKit Morgan Fingerprint similarity) and then every fourth pair chosen to give the dataset shown in Table 3. These examples hence provide pairs of molecules that exhibit a wide range of levels of structural similarity.

Each algorithm was run ten times on each of the pairs in Tables 2 and 3 in turn, using the MCS definitions that were appropriate for that algorithm as listed in Table 1; for example, VFLibMCS was run to identify just the cMCS whereas the Bron–Kerbosch runs additionally sought the dMCS and tdMCS. Runs exceeding 300 seconds were recorded as being incomplete. All reported MCS results are derived from the MCES (edge-induced), where atoms were matched based on atomic number, and bonds based on bond order (and matching atom types in both bonds).

Acknowledgements

E.D. was supported by a University of Sheffield studentship.

Conflict of interest

The authors declare no conflict of interest.

Keywords: chemoinformatics · drug discovery · maximum common subgraph · maximum common substructure · molecular alignment

[1] J. W. Raymond, P. Willett, *J. Comput.-Aided Mol. Des.* **2002**, *16*, 521–533.

[2] H. C. Ehrlich, M. Rarey, *Wiley Interdiscip. Rev.: Comput. Mol. Sci.* **2011**, *1*, 68–79.

[3] E. Duesbury, J. D. Holliday, P. Willett, *MATCH-Commun. Math. Comput. Chem.* **2017**, *77*, 213–232.

[4] D. Fooshee, A. Andronico, P. Baldi, *J. Chem. Inf. Model.* **2013**, *53*, 2812–2819.

[5] A. Böcker, *J. Chem. Inf. Model.* **2008**, *48*, 2097–2107.

[6] A. G. Leach, H. D. Jones, D. A. Cosgrove, P. W. Kenny, L. Ruston, P. MacFaul, J. M. Wood, N. Colclough, B. Law, *J. Med. Chem.* **2006**, *49*, 6672–6682.

[7] E. Griffen, A. G. Leach, G. R. Robb, D. J. Warner, *J. Med. Chem.* **2011**, *54*, 7739–7750.

[8] B. Zhang, M. Vogt, G. M. Maggiora, J. Bajorath, *J. Comput. Aided-Mol. Des.* **2015**, *29*, 937–950.

[9] T. Kawabata, H. Nakamura, *J. Chem. Inf. Model.* **2014**, *54*, 1850–1863.

[10] "The Complexity of Theorem-Proving Procedures", S. A. Cook, *STOC '71 Proceedings of the Third Annual ACM Symposium on Theory of Computing*, **1971**, pp. 151–158, http://doi.acm.org/10.1145/800157.805047 (accessed June 22, 2017).

[11] M. R. Garey, D. S. Johnson, *Computers and Intractability*, W. H. Freeman, San Francisco, **1979**.

[12] G. Levi, *CALCOLO* **1973**, *9*, 341–352.

[13] H. G. Barrow, R. M. Burstall, *Inf. Process. Lett.* **1976**, *4*, 83–84.

[14] Y. Takahashi, M. Sukekawa, S. Sasaki, *J. Chem. Inf. Comput. Sci.* **1992**, *32*, 639–643.

[15] S. Klinger, J. Austin J. *Weighted Superstructures for Chemical Similarity Searching*, **2006**, https://www.cs.york.ac.uk/arch/publications/byyear/2006/2006_WeightedSuperstructuresForChemicalSimilarity_234.pdf/at_download/2006_WeightedSuperstructuresForChemicalSimilarity_234.pdf (accessed June 22, 2017).

[16] E. J. Barker, D. Buttar, D. A. Cosgrove, E. J. Gardiner, P. Kitts, P. Willett, V. J. Gillet, *J. Chem. Inf. Model.* **2006**, *46*, 503–511.

[17] T. Kawabata, *J. Chem. Inf. Model.* **2011**, *51*, 1775–1787.

[18] P. Englert, P. Kovács, *J. Chem. Inf. Model.* **2015**, *55*, 941–955.

[19] J. W. Raymond, E. J. Gardiner, P. Willett, *J. Chem. Inf. Comput. Sci.* **2002**, *42*, 305–316.

[20] E. Duesbury, *Applications and Variations of the Maximum Common Subgraph for the Determination of Chemical Similarity*, PhD Thesis, University of Sheffield, **2015**, http://etheses.whiterose.ac.uk/view/creators/Duesbury=3AEdmund=3A=3A.default.html (accessed June 22, 2017).

[21] A. T. Brint, P. Willett, *J. Chem. Inf. Comput. Sci.* **1987**, *27*, 152–158.

[22] E. J. Gardiner, P. J. Artymiuk, P. Willett, *J. Mol. Graphics Modell.* **1997**, *15*, 245–253.

[23] S. A. Rahman, M. Bashton, G. L. Holliday, R. Schrader, J. M. Thornton, *J. Cheminf.* **2009**, *1*, 12.

[24] R. Hariharan, A. Janakiraman, R. Nilakantan, B. Singh, S. Varghese, G. Landrum, A. Schuffenhauer, *J. Chem. Inf. Model.* **2011**, *51*, 788–806.

[25] A. Dalke, *Varkony Reconsidered* **2013**, http://www.dalkescientific.com/writings/diary/archive/2013/07/27/varkony_reconsidered.html (accessed June 22, 2017).

[26] M. Depolli, J. Konc, K. Rozman, R. Trobec, D. Janežič, *J. Chem. Inf. Model.* **2013**, *53*, 2217–2228.

[27] C. Bron, J. Kerbosch, *Commun. ACM* **1973**, *16*, 575–577.

[28] R. Carraghan, P. M. Pardalos, *Oper. Res. Lett.* **1990**, *9*, 375–382.

[29] J. W. Raymond, E. J. Gardiner, P. Willett, *Comput. J.* **2002**, *45*, 631–644.

[30] Y. Zhu, L. Qin, J. X. Yu, Y. Ke, X. Lin, *VLDB J.* **2013**, *22*, 345–368.

[31] A. Grosso, M. Locatelli, W. Pullan, *J. Heuristics* **2008**, *14*, 587–612.

[32] A. Dalke, *Topologically Non-planar Compounds*, **2012**, http://dalkescientific.com/writings/diary/archive/2012/05/18/nonplanar_compounds.html (accessed June 22, 2017).

[33] D. S. Wishart, C. Knox, A. C. Guo, D. Cheng, S. Shrivastava, D. Tzur, B. Gautam, M. Hassanali, *Nucleic Acids Res.* **2008**, *36*, D901–D906.

[34] P. Franco, N. Porta, J. D. Holliday, P. Willett, *J. Cheminf.* **2014**, *6*, 5.

[35] K. Ersmark, I. Feierberg, S. Bjelic, J. Hultén, B. Samuelsson, J. Åqvist, A. Hallberg, *Bioorg. Med. Chem.* **2003**, *11*, 3723–3733.

[36] R. Bone, J. P. Vacca, P. S. Anderson, M. K. Holloway, *J. Am. Chem. Soc.* **1991**, *113*, 9382–9384.

[37] P. R. Libby, B. R. Munson, R. J. Fiel, C. W. Porter, *Biochem. Pharmacol.* **1995**, *50*, 1527–1530.

[38] R. M. Kariba, P. J. Houghton, A. Yenesew, *J. Nat. Prod.* **2002**, *65*, 566–569.

[39] C. Huang, Q. Xu, C. Chen, C. Song, Y. Xu, Y. Xiang, Y. Feng, H. Ouyang, Y. Zhang, H. Jiang, *J. Chromatogr. A* **2014**, *1361*, 139–152.

[40] L.-M. Li, G.-Y. Li, L.-S. Ding, C. Lei, L.-B. Yang, Y. Zhao, Z.-Y. Weng, S.-H. Li, S.-X. Huang, W.-L. Xiao, *Tetrahedron Lett.* **2007**, *48*, 9100–9103.

[41] P. J. Maurer, H. Rapoport, *J. Med. Chem.* **1987**, *30*, 2016–2026.

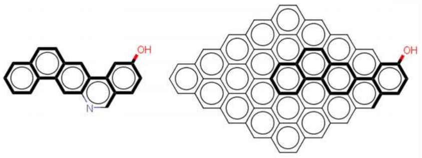
[42] M. R. Berthold, N. Cebron, F. Dill, T. R. Gabriel, T. Kötter, T. Meinl, P. Ohl, C. Sieb, K. Thiel, B. Wiswedel in *Data Analysis, Machine Learning and Applications*, Springer, Heidelberg, **2008**, pp. 319–326.■■correct city?■■

[43] C. Steinbeck, Y. Han, S. Kuhn, O. Horlacher, E. Luttmann, E. Willighagen, *J. Chem. Inf. Comput. Sci.* **2003**, *43*, 493–500.

[44] J. L. Hintze, R. D. Nelson, *Am. Stat.* **1998**, *52*, 181–184.

Manuscript received: August 13, 2017
Revised manuscript received: October 10, 2017
Accepted manuscript online: ■■ ■■, 0000
Version of record online: ■■ ■■, 0000

Finding the MCS (maximum common substructure) is a computationally difficult challenge, for which several attempts have been made to improve both the search speeds, and quality, of reported solutions. This article describes challenging benchmarks for a series of MCS algorithms, and reports the MCS type and algorithm combinations which generally yield the fastest and most sensible results in a chemoinformatic problem domain.



Quickest cMCS: 0.011 s
Quickest dMCS: 0.400 s

E. Duesbury, J. Holliday, P. Willett*

■■–■■

Comparison of Maximum Common Subgraph Isomorphism Algorithms for the Alignment of 2D Chemical Structures

#Chemoinformatics #DrugDiscovery @UCB news @SheffieldUni finds max. common substructure combos w/ rapid & best results SPACE RESERVED FOR IMAGE AND LINK

Share your work on social media! *ChemMedChem* has added Twitter as a means to promote your article. Twitter is an online microblogging service that enables its users to send and read text-based messages of up to 140 characters, known as “tweets”. Please check the pre-written tweet in the galley proofs for accuracy. Should you or your institute have a Twitter account, please let us know the appropriate username (i.e., @accountname), and we will do our best to include this information in the tweet. This tweet will be posted to the journal’s Twitter account @ChemMedChem (follow us!) upon online publication of your article, and we recommended you to repost (“retweet”) it to alert other researchers about your publication.

Please check that the ORCID identifiers listed below are correct. We encourage all authors to provide an ORCID identifier for each coauthor. ORCID is a registry that provides researchers with a unique digital identifier. Some funding agencies recommend or even require the inclusion of ORCID IDs in all published articles, and authors should consult their funding agency guidelines for details. Registration is easy and free; for further information, see http://orcid.org/.

Dr. Edmund Duesbury http://orcid.org/0000-0001-8394-6835
Dr. John Holliday
Prof. Dr. Peter Willett