BIOINFORMATICS APPLICATIONS NOTE

Vol. 29 no. 21 2013, pages 2792–2794
doi:10.1093/bioinformatics/btt475

Structural bioinformatics

Advance Access publication August 20, 2013

fmcsR: mismatch tolerant maximum common substructure searching in R

Yan Wang, Tyler W. H. Backman, Kevin Horan and Thomas Girke*
Department of Botany and Plant Sciences, University of California, Riverside, CA 92521, USA

Associate Editor: Anna Tramontano

ABSTRACT

Motivation: The ability to accurately measure structural similarities among small molecules is important for many analysis routines in drug discovery and chemical genomics. Algorithms used for this purpose include fragment-based fingerprint and graph-based maximum common substructure (MCS) methods. MCS approaches provide one of the most accurate similarity measures. However, their rigid matching policies limit them to the identification of perfect MCSs. To eliminate this restriction, we introduce a new mismatch tolerant search method for identifying flexible MCSs (FMCSs) containing a user-definable number of atom and/or bond mismatches.

Results: The *fmcsR* package provides an R interface, with the time-consuming steps of the FMCS algorithm implemented in C++. It includes utilities for pairwise compound comparisons, structure similarity searching, clustering and visualization of MCSs. In comparison with an existing MCS tool, fmcsR shows better time performance over a wide range of compound sizes. When mismatching of atoms or bonds is turned on, the compute times increase as expected, and the resulting FMCSs are often substantially larger than their strict MCS counterparts. Based on extensive virtual screening (VS) tests, the flexible matching feature enhances the enrichment of active structures at the top of MCS-based similarity search results. With respect to overall and early enrichment performance, FMCS outperforms most of the seven other VS methods considered in these tests.

Availability: fmcsR is freely available for all common operating systems from the Bioconductor site (http://www.bioconductor.org/packages/devel/bioc/html/fmcsR.html).

Contact: thomas.girke@ucr.edu

Supplementary information: Supplementary data are available at *Bioinformatics* online.

Received on March 14, 2013; revised on July 24, 2013; accepted on August 12, 2013

1 INTRODUCTION

The development of computational methods for detecting and quantifying structural similarities among small molecules is an area of intensive research in drug discovery and chemical genomics. The effort is largely driven by the observation that many structurally related compounds share similar bioactivity and physicochemical properties (Wale *et al*., 2010). Maximum common substructure (MCS) approaches are commonly used to identify the largest substructure (subgraph) shared among two compounds (Cao *et al*., 2008a; Conte *et al*., 2004; Hariharan *et al*., 2011; Hattori *et al*., 2003; Rahman *et al*., 2009; Raymond and Willett, 2002; Wang *et al*., 2009). MCS is a pairwise graph matching concept that differs fundamentally from the structural descriptor-based methods, but its results (e.g. size of MCS relative to source structures) can be used for the computation of related similarity coefficients. Compared with descriptor-based similarity methods, MCS approaches generate chemically more meaningful search results by pinpointing the common features within the structure of two compounds. They also provide the most efficient method for identifying local structural similarities and similarities among compounds with large size differences (Rahman *et al*., 2009). With the exception of bond mismatches, existing MCS search algorithms can only identify MCSs that are perfect substructure matches in two compounds of interest. Extending this strict matching scheme to one that tolerates mismatches among atoms and/or bonds facilitates the identification of larger flexible MCSs (FMCSs) than their strict MCS counterparts, resulting in a more complete description of the similarities among two compounds. Here we introduce such a FMCS algorithm that allows both mismatches of atoms and/or bonds in the identified MCSs. When the flexible matching feature is enabled, the user can identify more complex and subtle similarity patterns among two structures than is possible with strict MCS algorithms (Fig. 1A). For instance, two molecules may share a larger imperfect MCS that is disrupted by a substitution of a small number of atoms. Strict MCS algorithms will identify in these cases only the largest invariant subcomponent(s), while the FMCS algorithm will often find a much larger, but partially imperfect MCS (Fig. 1A). This has various advantages for practical applications in small molecule discovery, such as the prediction of bioactive compounds, scaffold identification in screening libraries or assignment of metabolic compounds to enzymatic steps in pathways.

2 METHODS

2.1 Background

To meet the above requirements, we designed the FMCS algorithm as an extension of our previously published backtracking VF algorithm for MCS detection (Cao *et al*., 2008a) where we enabled flexible matching by introducing counters for bond and atom mismatch tracking, and further optimized its time performance. A detailed outline of the algorithm is provided in the Supplementary Materials Section S1. To achieve both optimal performance and usability of the method for data mining applications, the time-consuming computational steps of *fmcsR* are implemented in C++, and its R interface integrates the small molecule analysis utilities and S4 object classes provided by the *ChemmineR* library (Backman *et al*., 2011; Cao *et al*., 2008b; O'Boyle *et al*., 2008).

*To whom correspondence should be addressed.

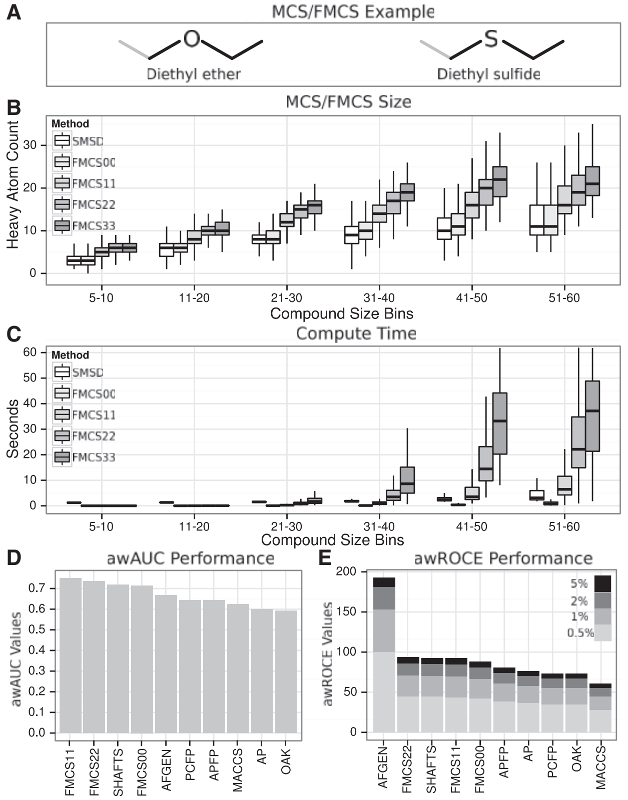

Fig. 1. Performance comparisons of FMCS with other methods. (**A**) The MCS and FMCS shared among two small molecules are highlighted. The gray fragment is the MCS result, and the gray and black fragments combined are the FMCS result obtained with one atom mismatch. (**B**) The plot compares the size distributions of MCS/FMCS results. Test compound pairs were randomly selected from DrugBank considering 1000 pairs within each of six size categories (*x*-axis) ranging from 5 to 60 non-hydrogen atoms. Each member of a compound pair had to fall into the same size category. The size distributions of the MCS/FMCS matches (*y*-axis) computed for these pairs are represented as box plots. The SMSD results are given in white and the FMCS results with three different mismatch settings are given in gray allowing zero (FMCS00), one (FMCS11), two (FMCS22) or three (FMCS33) bond and atom mismatches. (**C**) The corresponding compute time distributions of the four methods are plotted for the same dataset used in the previous plot. (**D**) The VS performance of FMCS with three different mismatch settings is compared with seven other methods (atom pairs, AP; atom pair fingerprints, APFP; PubChem fingerprints, PCFP; MACCS keys; the graph fragment-based method AFGen, the 3D methods SHAFTS and OAK). Averaged awAUC (arithmetically weighted area under the receiver-operating characteristic curve) values are plotted and the methods have been sorted along the x-axis by their performance. As test dataset, a subset of 13 compound sets from the Directory of Useful Decoys was used that is optimized for benchmarking VS experiments. More details and source data are provided in the Supplementary Materials S2 and Supplementary Table S1. (**E**) The plot compares the early enrichment performance of the same methods and test datasets used in the previous plot. The results are plotted as awROCE (arithmetically weighted receiver-operating characteristic enrichment) values obtained at false-positive rates of 0.5, 1.0, 2 and 5% (for details see Supplementary Materials S2 and Supplementary Table S6)

2.2 Main functionalities of *fmcsR*

The FMCS algorithm can be called in R from the *fmcs* function, which computes either the MCS or FMCS shared among two compounds and returns the result as an object of class S4 containing one or many alternative solutions. The speed-optimized *fmcsBatch* function provides MCS/FMCS-based search functionalities of small molecule databases. The number of allowable atom/bond mismatches and ring matching policies are user-definable parameters. A plotting function is available to visualize MCS/FMCS results by color highlighting the corresponding bonds in their source structures. A detailed user manual is included in the *fmcsR* package.

3 RESULTS

Figure 1B and C compare the performance of *fmcsR* with the MCS algorithm implemented in the SMSD toolkit (Rahman *et al.*, 2009). With perfect matching (FMCS00 in Fig. 1B), the FMCS algorithm returns MCSs with similar size distributions as SMSD, but with shorter compute times (Fig. 1C). Slight differences in the MCS size distributions among the two methods are as expected owing to (i) differences in their perception and matching behavior of rings and aromatic bonds, (ii) different heuristics used for improving time performance and (iii) different MCS concepts considered by the two methods (for details see Supplementary Materials S1; Cao *et al.*, 2008a; Rahman *et al.*, 2009). For extremely large compounds, above 60 non-hydrogen atoms, SMSD switches to a faster approximation approach where it exhibits a better time performance than FMCS in its perfect matching mode. When allowing 1, 2 or 3 mismatches of bonds and atoms, the size distributions of the FMCS results increase on average by 20–50% compared with the strict MCS results. On average these size increases substantially exceed the number of allowed mismatches because a mismatch will often allow the algorithm to identify many additional matching bonds and atoms resulting in an FMCS of a much larger size than the corresponding MCS. For instance, the MCS shared among diethyl ether and diethyl sulfide contains only two non-hydrogen atoms, but their FMCS with one atom mismatch contains five non-hydrogen atoms (Fig. 1A). With relaxed mismatch parameters, the complexity of the FMCS computation increases, resulting in longer processing times. Nevertheless, for 1–2 atom and bond mismatches the compute times are still acceptable across the six compound size bins considered in Figure 1C. In virtual screening (VS) benchmark tests, the FMCS method shows consistently better performance than strict MCS matching (Fig. 1D and E), indicating that the mismatch tolerant MCS similarity concept improves the early and late enrichment performance of MCS-based search methods in VS experiments (Good and Oprea, 2008; Huang *et al.*, 2006). The strong overall performance of FMCS, compared with a diverse set of seven 2D and 3D structure similarity search algorithms (Chen and Reynolds, 2002; Jahn *et al.*, 2009; Liu *et al.*, 2011; Wale *et al.*, 2010), demonstrates its usefulness for this application field. The details for these extensive VS performance tests are provided in the Supplementary Materials S2.

4 CONCLUSIONS

The *fmcsR* package introduces a versatile algorithm for identifying both MCSs and FMCSs. Its mismatch tolerant matching mode provides a more complete description of subtle similarity patterns shared among compounds than this is possible with strict MCS detection methods.

ACKNOWLEDGEMENT

The authors acknowledge the support of the compute facility at the Institute for Integrative Genome Biology (IIGB) at UC Riverside.

Funding: National Science Foundation (ABI-0957099 and IGERT-0504249 to T.G.).

Conflict of Interest: none declared.

REFERENCES

Backman,T.W. *et al.* (2011) ChemMine tools: an online service for analyzing and clustering small molecules. *Nucleic Acids Res.*, **39**, 486–491.

Cao,Y. *et al.* (2008a) A maximum common substructure-based algorithm for searching and predicting drug-like compounds. *Bioinformatics*, **24**, 366–374.

Cao,Y. *et al.* (2008b) ChemmineR: a compound mining framework for R. *Bioinformatics*, **24**, 1733–1734.

Chen,X. and Reynolds,C.H. (2002) Performance of similarity measures in 2D fragment-based similarity searching: comparison of structural descriptors and similarity coefficients. *J. Chem. Inf. Comput. Sci.*, **42**, 1407–1414.

Conte,D. *et al.* (2004) Thirty years of graph matching in pattern recognition. *Int. J. Pattern Recognit. Artif. Intell.*, **18**, 265–298.

Good,A.C. and Oprea,T.I. (2008) Optimization of CAMD techniques 3. Virtual screening enrichment studies: a help or hindrance in tool selection? *J. Comput. Aided Mol. Des.*, **22**, 169–178.

Hariharan,R. *et al.* (2011) MultiMCS: a fast algorithm for the maximum common substructure problem on multiple molecules. *J. Chem. Inf. Model.*, **51**, 788–806.

Hattori,M. *et al.* (2003) Heuristics for chemical compound matching. *Genome Inform.*, **14**, 144–153.

Huang,N. *et al.* (2006) Benchmarking sets for molecular docking. *J. Med. Chem.*, **49**, 6789–6801.

Jahn,A. *et al.* (2009) Optimal assignment methods for ligand-based virtual screening. *J. Cheminform.*, **1**, 14.

Liu,X. *et al.* (2011) SHAFTS: a hybrid approach for 3D molecular similarity calculation. 1. Method and assessment of virtual screening. *J. Chem. Inf. Model.*, **51**, 2372–2385.

O'Boyle,N.M. *et al.* (2008) Pybel: a Python wrapper for the OpenBabel cheminformatics toolkit. *Chem. Cent. J.*, **2**, 5.

Rahman,S.A. *et al.* (2009) Small Molecule Subgraph Detector (SMSD) toolkitl. *J. Cheminform.*, **1**, 12.

Raymond,J.W. and Willett,P. (2002) Maximum common subgraph isomorphism algorithms for the matching of chemical structures. *J. Comput. Aided Mol. Des.*, **16**, 521–533.

Wale,N. *et al.* (2010) Trends in chemical graph data mining. In: Aggarwal,C.C. and Wang,H. (eds) *Managing and Mining Graph Data*. Springer, New York, pp. 581–606.

Wang,X. *et al.* (2009) G-hash: towards fast kernel-based similarity search in large graph databases. In: *Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology*. ACM, Saint-Petersburg, Russia, pp. 472–480.