Computationally Efficient Algorithm to Identify Matched Molecular Pairs (MMPs) in Large Data Sets

Jameed Hussain* and Ceara Rea

Computational & Structural Chemistry, GlaxoSmithKline, Medicines Research Centre, Gunnels Wood Road, Stevenage, Hertfordshire, SG1 2NY, U.K.

Received November 19, 2009

Modern drug discovery organizations generate large volumes of SAR data. A promising methodology that can be used to mine this chemical data to identify novel structure−activity relationships is the matched molecular pair (MMP) methodology. However, before the full potential of the MMP methodology can be utilized, a MMP identification method that is capable of identifying all MMPs in large chemical data sets on modest computational hardware is required. In this paper we report an algorithm that is capable of systematically generating all MMPs in chemical data sets. Additionally, the algorithm is computationally efficient enough to be applied on large data sets. As an example the algorithm was used to identify the MMPs in the ~300k NIH MLSMR set. The algorithm identified ~5.3 million matched molecular pairs in the set. These pairs cover ~2.6 million unique molecular transformations.

INTRODUCTION

Drug discovery is a multiobjective process; a compound needs to be optimized for a number of properties (including potency, bioavailability, in vivo activity, safety, etc.) before it can become a successful drug candidate. A typical scenario within drug discovery programs is the situation where the most promising lead compound found from screening needs to be further improved in one or more properties in the lead optimization process before it can be considered as a clinical candidate.

The fundamental process used within the lead optimization process to improve a compound's properties has not changed greatly over the years. The process involves the formulation of a hypothesis to explain the presence of a particular liability followed by the synthesis of further compounds and experimental measurements to test the hypothesis. What has changed significantly (over the years), is the advancement of the various techniques involved in this process. Through the advent of parallel chemistry and improved screening technologies it is now possible to rapidly synthesize and biologically screen large numbers of compounds.

The processes used to generate a hypothesis on the structural moieties responsible for a particular liability still rely heavily on the collective experience of the medicinal chemistry team. On the basis of the experience of the team, a chemical substitution can be suggested that has worked in the literature or other medicinal chemistry projects. However, given the large amount of data available in drug discovery organizations, it is likely that a particular medicinal chemistry team will not be aware of a (possibly large) number of chemical substitutions that have been successfully used in the past to remove a particular liability.[1−3] In silico global QSAR models, which are now available for a number of properties, provide a way of capturing the knowledge available within these large data sets. However, most of these models are used to score or prioritize existing compound suggestions; very few can be used to suggest chemical changes to make to a compound to improve its properties (i.e., inverse QSAR[2]). To develop a system that is capable of suggesting chemical changes to improve a particular property, one first needs to devise an automated and systematic methodology to mine existing experimental data for chemical substitutions that lead to an improvement. A promising approach that can be used for this purpose is the matched molecular pair (MMP) methodology.[4]

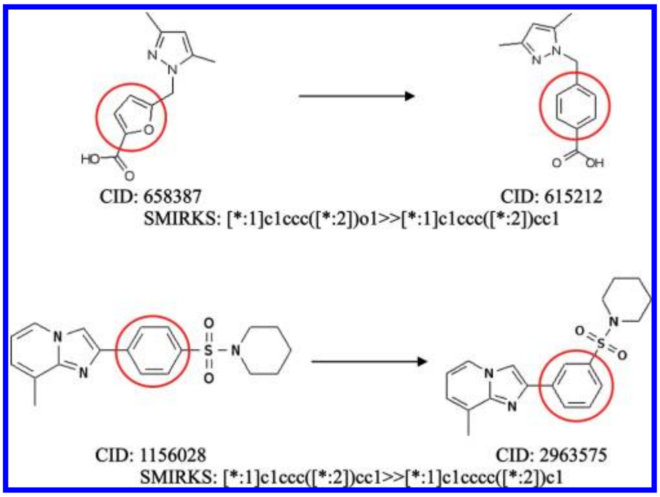

HO
O
O
N
N
O
OH
CID: 658387
CID: 615212
SMIRKS: [*:1]c1ccc([*:2])o1>>[*:1]c1ccc([*:2])cc1
N
N
O
S
O
N
N
N
O
S
O
N
CID: 1156028
CID: 2963575
SMIRKS: [*:1]c1ccc([*:2])cc1>>[*:1]c1cccc([*:2])c1

Figure 1. Two examples of MMPs with SMIRKS describing the transformation. Note: CID refers to the PubChem Compound ID.

A MMP is a pair of compounds (compound A and compound B) that only differ by a single localized structural change[4] (see Figure 1). Therefore, the compounds belonging to an MMP can be converted to one another (i.e., compound A to compound B) by the molecular transformation of substructure A to substructure B (where substructures A and B are the substructures that have changed from compound

* To whom correspondence should be addressed. E-mail: jameed.x.hussain@gsk.com.

10.1021/ci900450m
Published on Web 02/01/2010

A to compound B; see circled substructures in Figure 1). Another parameter that needs to be defined for MMPs is the maximum size of substructures A and B. The larger this parameter, the greater the number of valid MMPs in a data set. As MMPs typically only deal with single point changes, substructures A and B need to be single nondisconnected substructures otherwise the MMPs would consist of more than one change. Studying the effect of these defined structural changes (across multiple MMPs) on measurements in relevant assays (e.g., solubility, clearance, CNS penetration, etc.) can lead to the discovery of trends that can form the basis of new structure−activity relationships. Another example of the use of MMPs is in novel bioisostere identification, that is, finding pairs of compounds where a chemical group has been replaced and the biological activity is retained. The MMP methodology has successfully been utilized in the literature to study bioisosterism,[3,5,6] aqueous solubility,[2,7,8] plasma protein binding,[8,2] oral exposure,[8] local SAR,[9] ligand potency,[10,11] intrinsic clearance,[12] hERG and P450 metabolism.[7]

MMP Identification. The goal of the MMP data mining concept is the identification of novel structure−activity relationships; therefore, an ideal MMP identification algorithm needs to satisfy the following two requirements: (1) be capable of identifying all matched molecular pairs in a data set and (2) be computationally efficient so the algorithm can be applied to large compound data sets on modest hardware.

To the best of our knowledge an MMP identification algorithm that satisfies both requirements has not yet been reported. A number of the reported methods to identify MMPs require the predefinition of the structural changes before its associated MMPs can be identified within a compound data set.[2,8,11,12] This includes the LEATHERFACE and find_pairs programs reported by Leach et al.,[8] where a molecular transformation is first defined using a SMARTS.[13] This transformation is then applied to a set of compounds and the resulting molecules are then matched back against the original compound set. This allows the identification of MMPs that differ by the defined transformation. A similar algorithm is used by Lewis et al.[12] The predefinition of the structural change is obviously not ideal because it makes the identification of novel (and hence unknown) structure−activity relationships difficult.

The MMP identification approach described by Haubertin et al.[2] can be considered more general. Here, a set of 9038 side chains were identified by applying the RECAP fragmentation algorithm[14] on their corporate database of compounds. The algorithm then takes a set of compounds, performs the RECAP fragmentation, and identifies compounds that contain any of the 9038 defined side chains. Once this is done, the identification of MMPs, which involve the substitution of any of the defined side chains is straightforward. The method could equally be applied to find MMPs that involve a transformation in compound "cores" (as well as side chains), but this has not been reported. The method offers an advantage over the LEATHERFACE or find_pairs programs because of the greater variety of chemical transformations and hence MMPs that can be identified. However, the method does not satisfy requirement 1 (above) because any MMPs containing transformations outside the defined side chain list will not be found.

The MMP identification method reported by Gleeson et al.[7] only requires a partial definition of the molecular transformation to identify relevant MMPs. Here, one needs to specify a substructure (substructure x) and the algorithm will find all the MMPs in a data set that involves a transformation of substructure x. The method works by performing multiple substructures searches. First, all compounds that contain substructure x are identified. The next step of the algorithm involves the removal of substructure x from one of the compounds identified (compound n_i). The resulting structure (compound n_i − substructure x) is then used as a substructure query and any compounds that are found from this search (compound m_i) are MMPs with compound n_i (with the molecular transformation, substructure $x \gg$ substructure y, where substructure y = compound m_i − (compound n_i − substructure x)). This is repeated for all compounds that contain substructure x. The method can be used to identify MMPs, where substructure x is a core or a terminal group. However, the algorithm does not satisfy requirement 1 (above) as only MMPs involving a transformation of substructure x will be found. Additionally, the method is likely to be computationally prohibitive on large data sets in cases where many common substructures are specified.

The algorithm reported by Hajduk et al.[10] provides a way to circumvent the limitation of requiring a predefined list of structural changes. Here, a pairwise comparison between all the compounds in a data set is performed using the findsubs routine available from Daylight.[13] The algorithm is limited to terminal or side group changes only. Additionally, because of the pairwise comparisons that need to be performed the computational efficiency of the algorithm is $O(n^2)$; hence (although the method was run on a set of ∼84k compounds), it is likely that the algorithm is unsuitable for very large data sets (∼1 million) because of the computational expense.

The final class of the reported MMP identification methods use maximum common subgraph (MCS) algorithms.[3,5,6,9] The algorithms essentially work by performing a pairwise comparison within a compound data set to find the MCS between each pair of compounds. The atoms within a pair of compounds that are not part of the MCS are then analyzed to determine if they constitute a single-point change.[3,5,9] Further restrictions can also be applied such as size constraints on the chemical change between a pair of compounds.[3] This final class of MMP identification algorithms is capable of potentially finding all MMPs within a compound data set, but because of the computation expense of MCS algorithms coupled with the $O(n^2)$ nature of the MMP identification (because of the pairwise comparisons that need to be performed), the algorithms are computationally expensive. Therefore, to alleviate this limitation, heuristics are performed between pairs of compounds before the MCS or the MMP identification step is carried out (clustering[5] and topological similarity[9]), which may result in certain MMPs not being found. Alternatively, the calculation needs to be performed on sufficiently powerful computational hardware. For example, in ref 3, where an MCS based algorithm was used to determine the MMPs in a 2.7 million compound data set, a 1072 CPU core cluster was utilized. The high computational cost of running these MMP identification

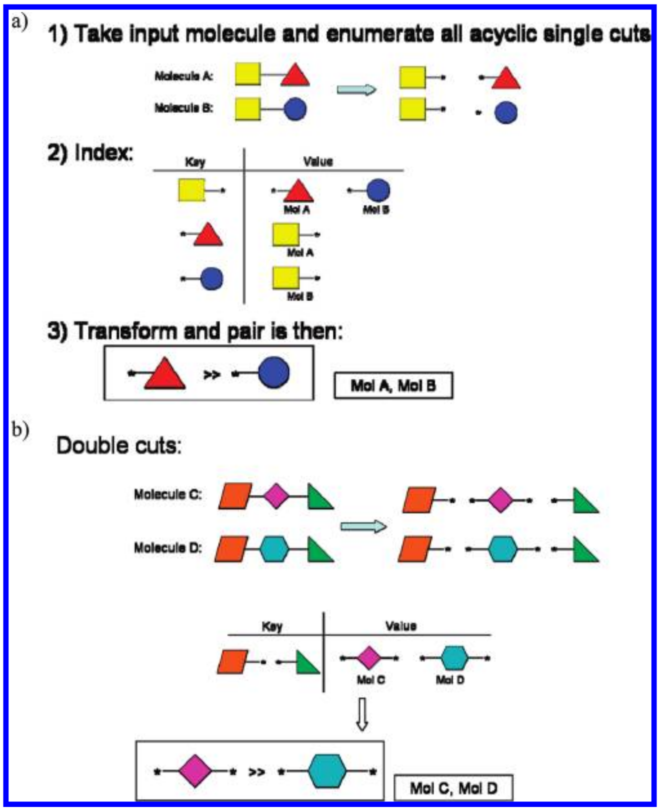


Figure 2. Schematic outline of the MMP identification algorithm. Panel a shows the steps involved when single cuts are performed, and panel b shows the equivalents steps involved when double cuts are performed.

algorithms means they are very difficult to apply on large compound data sets.

METHODS

In this paper, we report a new MMP identification algorithm that is capable of identifying all the MMPs within a data set. Additionally, it is computationally efficient enough to be applied to very large data sets on modest computational hardware. The algorithm is implemented using the Perl programming language[15] with the Daylight toolkit.[13] The program takes a set of SMILES[16] of the compounds under analysis as input and outputs the MMPs identified with a valid SMIRKS[13] describing the molecular transformation for each MMP.

The algorithm works by fragmenting and appropriately indexing the compounds under analysis. A schematic of the algorithm can be found in Figure 2. The first step of the algorithm is to fragment all the compounds in the input data set. For the sake of clarity, we will focus on the single cut example in Figure 2a. Here, each compound is examined and every acyclic single bond between two non-hydrogen atoms is marked.[17] The fragmentation then proceeds by enumerating all the possible single cuts (at the marked bonds) in a compound. An example of the fragments formed for a compound can be seen in Figure 3a.

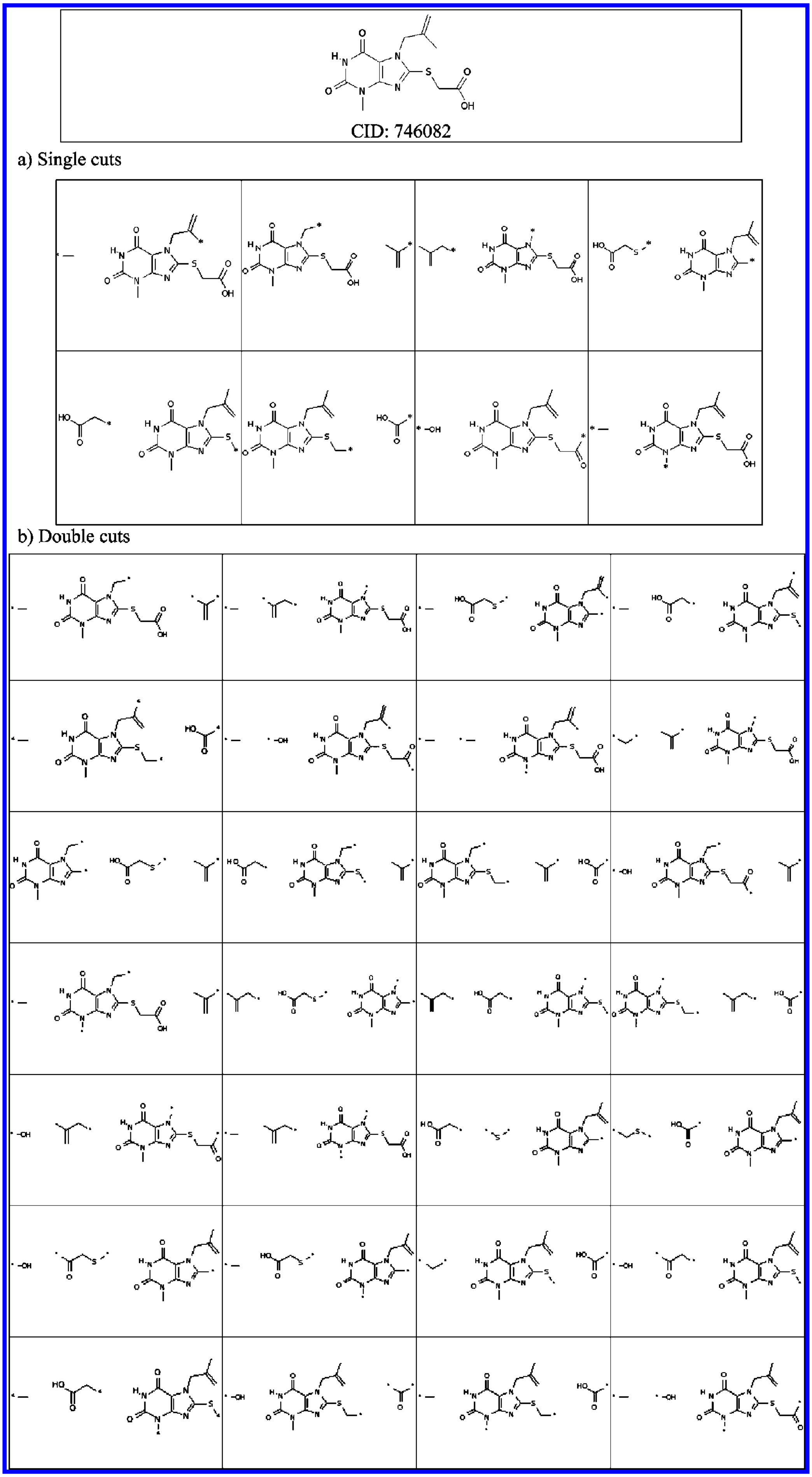
The next stage of the algorithm is to index these fragments. The structure of the index can be seen in Figure 2. When a single cut is made in an input compound, the two resulting fragments formed (i.e., fragment X and fragment Y) are both canonicalized and added to the index. First, fragment X is added as a “key” into the index with fragment Y as its “value”. The converse is also carried out; fragment Y is added as a key with fragment X as its value. Note, an identifier for the compound is also stored in the value of the index. This is repeated for all the possible fragmentations for a compound and on all compounds in the data set. Because of the canonicalization of the fragments, all the compounds belonging to a particular index key share the key fragment in their structure. Therefore, every pairwise combination of the compounds in the values of that particular key are valid MMPs (with the SMIRKS of the transformation being fragment1_in_value ≫ fragment2_in_value).

In addition to enumerating every single cut in a compound, double and triple cuts are also performed (Figure 2). In the double cut scenario, the fragmentations result in a core and two terminal fragments (see double cut “fragmentation” examples in Figure 3b). The core is stored as the value with the canonicalized dot-disconnected SMILES of the terminal fragments as their key. Similarly, in the triple cut example, only fragmentations that result in a core and three terminal groups are stored in the index (see valid triple cut examples in Figure 3c). As in the single cut example, all pairwise combinations of the compounds with the same key are MMPs where the cores have been replaced. On generation of the SMIRKS for these MMPs, the connectivity of the fragments with the core needs to be checked so the SMIRKS generated maintain the correct regiospecificity of each of the compounds in the MMP (see atom labels for the SMIRKS in Figure 1).

The index in the current form is not capable of finding MMPs with transformations that involve the substitution of a hydrogen atom (Figure 4). Therefore, an additional step needs to be performed so these MMPs can be found. Here, a hydrogen atom is added to the free position of the fragments (labeled with * in the fragmentations shown in Figure 3) in the key fragments of the index. This is only required for the single cut “fragmentations”. The resulting SMILES is canonicalized and compared with the canonicalized SMILES of the input structures. If a match is found, it is added as a value for that key with the SMILES “*[H]” as the fragment. This ensures MMPs involving a substitution of a hydrogen atom are found.

The algorithm is capable of finding MMPs irrespective of the size of the structural moiety that is modified. However, MMPs containing large substitutions are less interesting; therefore, the algorithm has a user specified parameter to remove these. The default value of this parameter is set to 10 non-hydrogen atoms, which means MMPs where the structural change involves the substitution of a structural moiety greater than 10 non-hydrogen atoms are not found. This is simply achieved by not adding the offending fragmentations to the index.

The molecular transformation or SMIRKS for a unique MMP in a data set can be represented in multiple ways. First, the direction of the MMP (ie. compound A to compound B or compound B to compound A) leads to two different SMIRKS. The program can be set to output the SMIRKS of a unique MMP in both directions. Second, the size of the substructure that changes from compound A to compound B can be represented in several ways (see Figure 5). Transformations represented by substructures of the minimum change (NH_2 to OH for both MMPs in Figure 5) to substructures of the maximum size (i.e., the whole structures)


CID: 746082
a) Single cuts
b) Double cuts

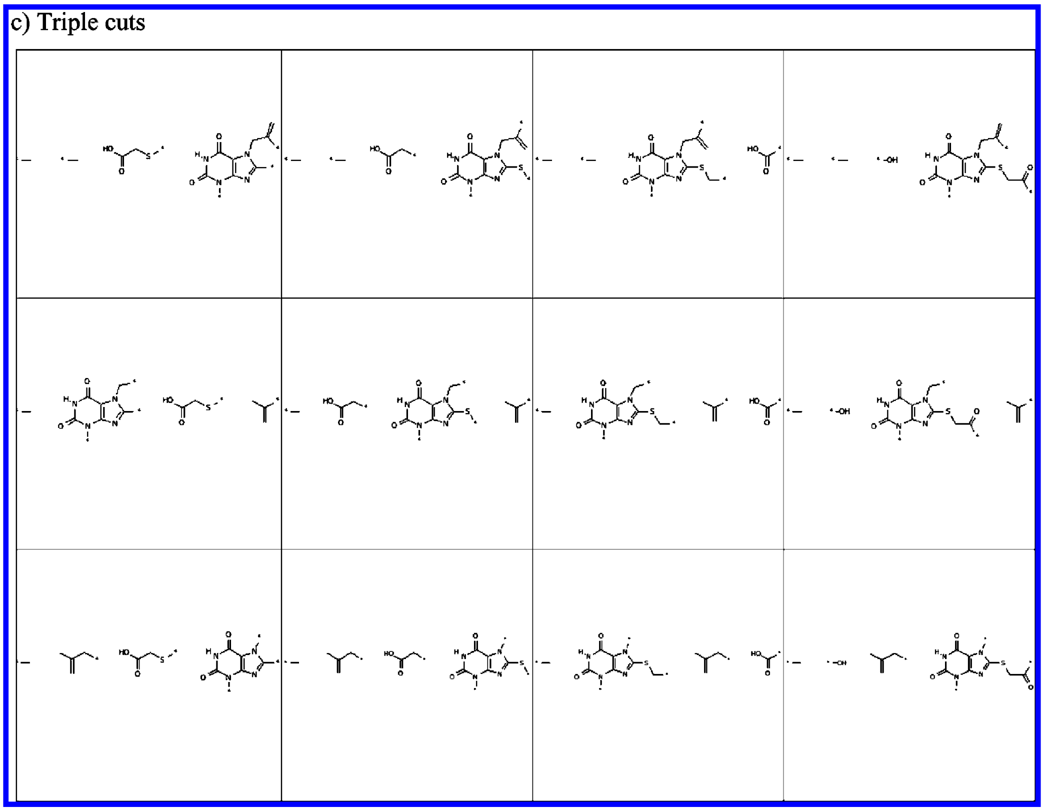


Figure 3. Example showing the fragments formed when enumerating all single cuts (a), all double cuts (b), and all triple cuts in compound 746082 (CID).

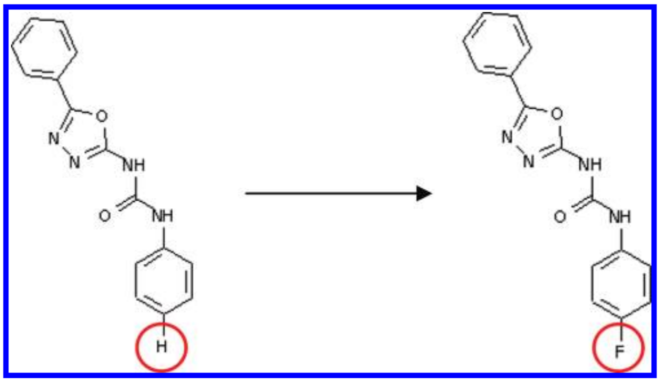


Figure 4. Example of an MMP where the molecular transformation involves the substitution of a hydrogen atom.

are all valid (in that they can be used to interconvert the compounds in an MMP). The algorithm will output all the SMIRKS that can be used to represent the MMPs as long as they are less than the user specified size (default set to 10 non-hydrogen atoms) and can be found by the fragmentation used (as only acyclic bonds are cut no SMIRKS involving the breakage of ring systems are output).

RESULTS

To test the MMP identification algorithm, it was applied to the NIH Molecular Libraries Small Molecule Repository[18] (MLSMR) set. The compound set was downloaded from the PubChem Web site[19] and contained 333 491 compounds. It was first filtered to remove compounds that are likely to be problematic in the algorithm[20] and compounds that were deemed to be highly undesirable for medicinal chemistry.[21] After the filtering process, 333 332 compounds remained, and this set was used to test the MMP identification algorithm.

The first part of the algorithm is the fragmentation step. This step can be run on multiple CPU cores; however, for this analysis, only a single CPU core was utilized. The fragmentation of the MLSMR set took 501 min.[22] The total number of fragmentations generated was 21 721 113, which represent an average of 65.2 unique fragmentations per compound. The final part of the program which performs indexing and identifies the MMPs (with the size parameter set to the default value of 10) took 356 min to complete on a single CPU core.[22] The number of unique MMPs identified was 5 310 964 with 2 585 772 unique transforms (ignoring the direction of the SMIRKS and finding the smallest SMIRKS for each MMP).

DISCUSSION

The twenty most frequently occurring molecular transformations for the MMPs found in the MLSMR data set are shown in Figure 6. The most frequently occurring transformation was hydrogen to methyl (45 717 MMPs). The most popular transformations identified are in accordance with what one would expect with the only surprise (to our eyes) being regiospecific transformations (the sixth and eighth most frequently occurring transformation with 5860 and 5425 MMPs respectively in Figure 6). The molecular transformations represent the smallest SMIRKS found for a unique MMP (where the direction of the change is ignored).

The most frequently occurring transformations that involve terminal group changes (found from the single cut fragmentations) and the core changes (found from the double and triple cut fragmentations) are shown in Figures 7−9, respectively. The most popular core change is a 1,4-substitution on a benzene ring to a 1,2-substitution (double

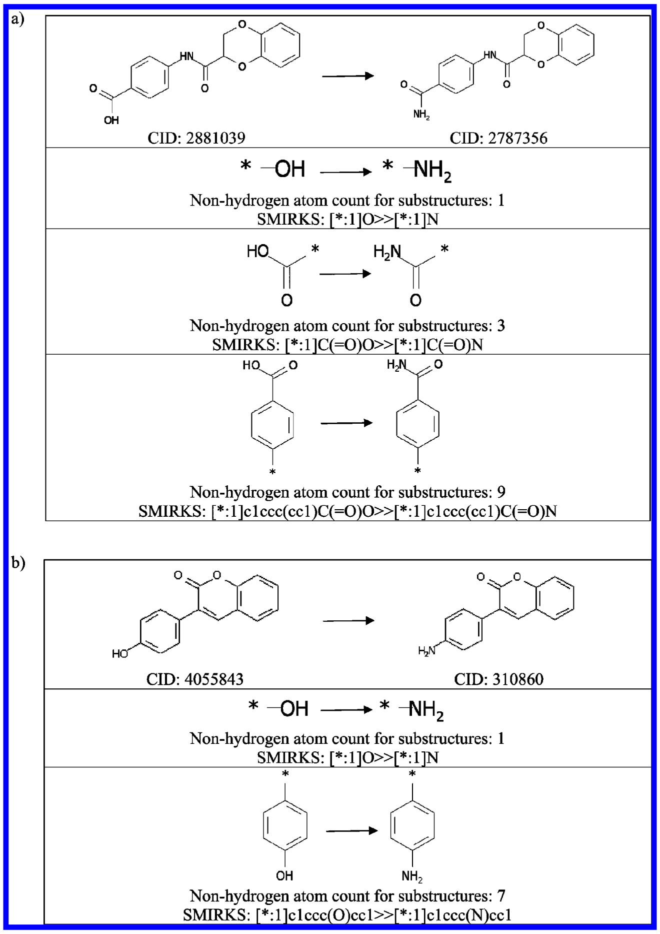

a)
CID: 2881039
CID: 2787356
* -OH → * -NH2
Non-hydrogen atom count for substructures: 1
SMIRKS: [*:1]O>>[*:1]N
Non-hydrogen atom count for substructures: 3
SMIRKS: [*:1]C(=O)O>>[*:1]C(=O)N
Non-hydrogen atom count for substructures: 9
SMIRKS: [*:1]c1ccc(cc1)C(=O)O>>[*:1]c1ccc(cc1)C(=O)N
b)
CID: 4055843
CID: 310860
* -OH → * -NH2
Non-hydrogen atom count for substructures: 1
SMIRKS: [*:1]O>>[*:1]N
Non-hydrogen atom count for substructures: 7
SMIRKS: [*:1]c1ccc(O)cc1>>[*:1]c1ccc(N)cc1

Figure 5. Example for the SMIRKS generated by the MMP identification algorithm to describe the molecular transformation for two different MMPs. The non-hydrogen atom count for the substructures is also shown.

cut) and a 1,3,6-substitution on a benzene ring to a 1,2,6-substitution (triplet cut). The MMPs that belong to these transformations are regioisomers.

The distribution of the molecular transformations against the count of the number of MMPs for the MLSMR data set (for the 2000 most frequently occurring transformations) is shown in Figure 10. As can be seen, the data has a Zipfian distribution,[23] with the hundred most frequently occurring transformations covering only 4.7% of the total MMPs found. Of the 2 585 772 unique transformations found 78.2% (2 020 927 transformations) only occur in one MMP.

The 2.5 million transforms found refer to the smallest SMIRKS (in terms of non-hydrogen atoms) for each unique MMP. They can be considered the minimum substructure change for an MMP. Note, that as no rings are broken in the fragmentation step, the minimum transformation found for MMPs that contain a change in a ring atom (Figure 11) are not strictly the minimum substructure change. The minimum substructure change in those cases is the substitution of the relevant atoms in the ring.

The algorithm outputs a number of SMIRKS for every unique MMP. The extra SMIRKS generated could be considered redundant when one is only interested in the unique MMPs and their minimum substructure change. This is the case for applications such as similarity searching (i.e., to determine the MMPs for a given compound in a corporate database to select a number of compounds for biological screening to generate SAR around the given compound). However, in data-mining applications where the objective is to determine novel structure−activity relationships, the extra SMIRKS for the MMPs can provide valuable context on the nature of the structural change. For example, the minimum substructure change for a MMP that involves the substitution of a carboxylic acid to an amide group (Figure 5a) and the substitution of an alcohol to an amine group (Figure 5b) will be the same (OH to NH_2). However, the change in compound properties for these MMPs is likely to be different (e.g., pK_a). Ideally, one would like to compare MMPs that involve the change of a carboxylic acid to an amide group in a separate analysis to a change that involves an alcohol group to an amine group; the multiple SMIRKS generated for a unique MMP means that this is possible. For example, the MMPs involving a carboxylic acid to amide group change can be analyzed by only selecting the MMPs with the SMIRKS that encode that change (“[*:1]C(=O)O≫[*:1]C(=O)N”). The analysis of MMPs involving an alcohol to an amine group substitution is more challenging and requires an additional step as the SMIRKS for that transform (“[*:1]O≫[*:1]N”) also retrieves MMPs where a carboxylic acid has been substituted with an amide (Figure 5). These MMPs need to be removed to yield a set of homogeneous MMPs (i.e., consisting of compounds with only alcohol to amine group changes) before an analysis can be performed. This additional step is required because the fragmentation step of the algorithm involves the cleavage of all acyclic single bonds. Therefore, an elegant way to remove the need for the additional step is to employ a more discriminatory fragmentation. If acyclic single bonds are only cleaved if they occur between functional groups (and not within functional groups, for example, the C−OH bond in carboxylic acids), a particular SMIRKS generated by the algorithm will not contain MMPs involving the substitution of two different functional groups (e.g., alcohol to amine and carboxylic acid to amide). The improved fragmentation method has now been implementated (by changing the SMARTS expression used to mark the bonds to be cleaved in the fragmentation step[24]); however, it was not available when the analysis of the MLSMR data set was performed.

The fragmentation step of the algorithm only involves the breakage of the acyclic bonds in a compound. Therefore, it may be expected that MMPs that involve a small change in a large ring system (which is greater in size than the maximum size parameter used) will not be found by the current implementation of the algorithm. One way this limitation can be overcome is by setting the maximum size parameter higher. However, this will result in the number of valid MMPs in a data set to increase (perhaps considerably), leading to an increase in the computation cost involved. Additionally, a proportion of the MMPs found are likely to be uninteresting as they involve a large structural change.[25] A more elegant way to overcome the problem is to allow fragmentations (that would result in MMPs where the substitution is greater that maximum size parameter to be found) to be indexed if the value part of the fragmentation is a “pure” ring system (i.e., it contains no ring atom to nonring atom single bonds). The adaptation would allow the index to capture MMPs where the transformation involves a small change in a large ring system. This adaptation is not

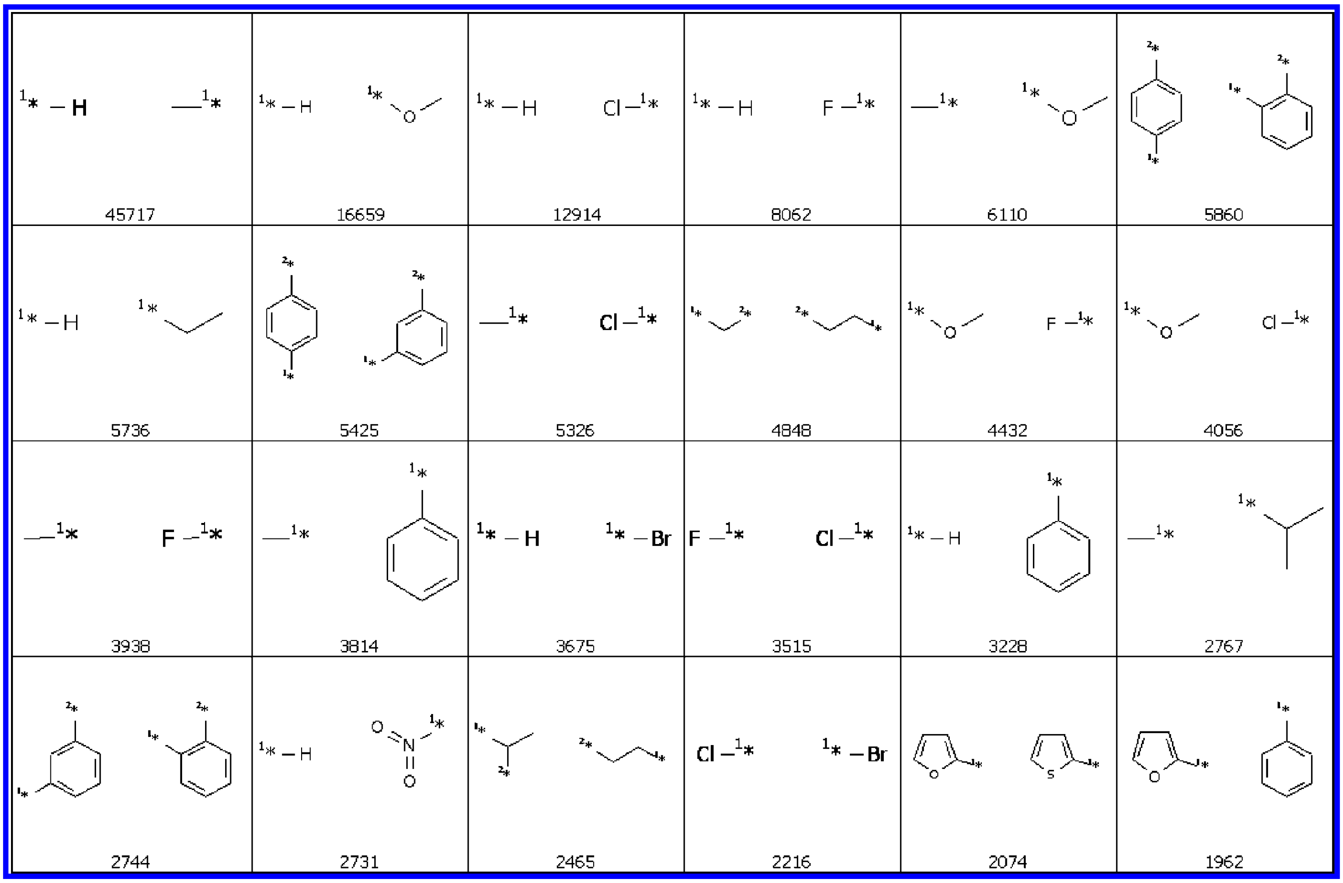


Figure 6. Most frequently occurring transformations found in the MLSMR data set. The number of MMPs found for each transformation is also shown.

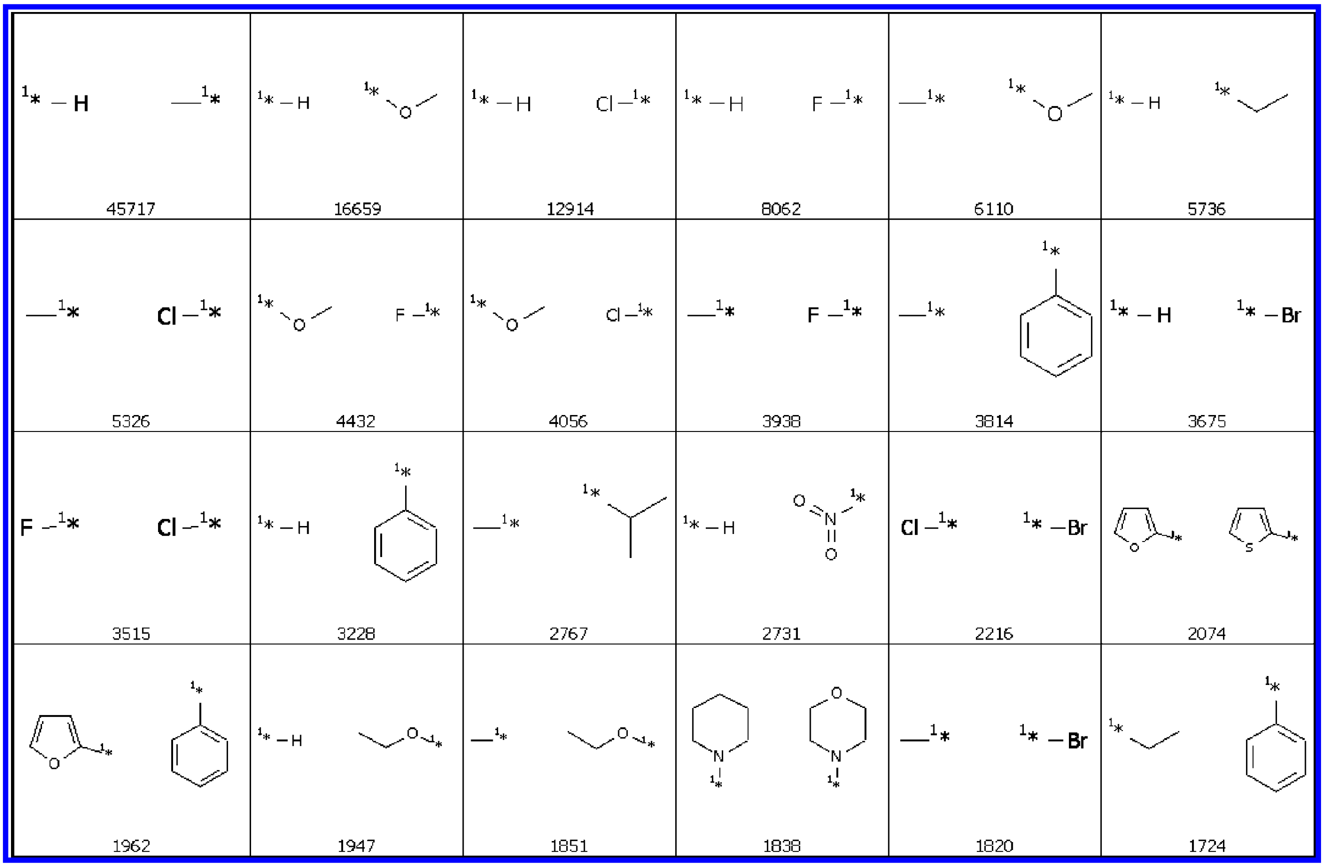


Figure 7. Most frequently occurring side chain/terminal transformations (found from the single cut fragmentations) in the MLSMR data set. The number of MMPs found for each transformation is also shown.

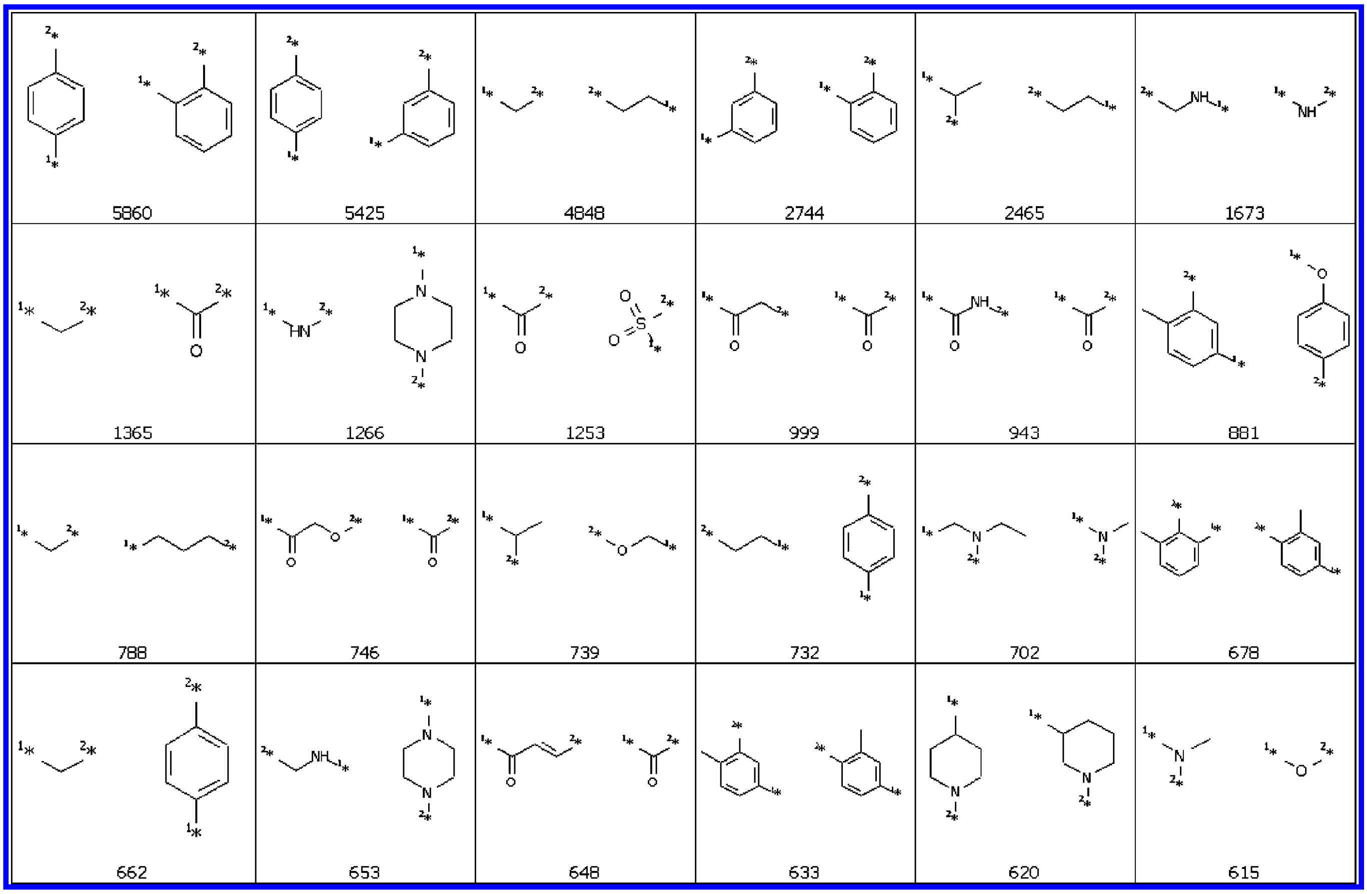


Figure 8. Most frequently occurring core transformations (found from the double cut fragmentations) in the MLSMR data set. The number of MMPs found for each transformation is also shown.

available in the current implementation; however, it is straightforward to do.

Another class of MMPs that the current implementation may not be able to identify is one which involves the substitution of a core that is connected to four side chains (i.e., quadruply substituted). The fragmentation only enumerates single, double and triple cuts in a compound: it would be straightforward to implement quadruple (or more) cuts in the fragmentation. However, in our opinion, it is only likely to result in a small increase in the number of unique MMPs found. To illustrate this, the algorithm was run with only single cuts and with single and double cuts and the number of unique MMP identified was determined. The results are shown in Table 1. As can be seen the number of MMPs identified from just the single cuts represents 80.2% of the total MMPs found (from applying single, double, and triplet cuts). The double cuts contribute a further 19.4% of the total MMPs identified. However, the triplet cuts only contribute a further 0.4% of the total MMPs found. We expect the laws of diminishing returns to apply and the extra MMPs identified for quadruple (or more) cuts to be smaller that the 0.4% increase seen on going from double to triple cuts. Therefore, although the current implementation does not completely satisfy requirement 1 of an ideal MMP identification algorithm, we believe the vast majority of MMPs can be found by enumerating single, double, and triple cuts only. Additionally, if quadruple (and more) cuts is implemented, the algorithm should be capable of identifying all the MMPs in a data set. In our opinion, the small increase in the number of MMPs found by implementing quadruple (or more) cuts is likely to be out-weighed by the greater computational cost that would result in the fragmentation step.

A parallel implementation of the fragmentation step has been implemented within GSK. This part of the algorithm scales linearly (O(n)) to the number of compounds in a data set under analysis. The second part of the algorithm which involves the indexing and the MMP identification, scales depending on the number of MMPs that are in a data set (at worse this is O(n^2) if the data set set is fully connected (i.e., every compound in the data set is an MMP with every other compound in the data set)). We have yet to encounter a fully connected data set and the number of MMPs found from a data set is typically much lower than the theoretical maximum. Note that for this analysis the time taken to perform the fragmentation step was longer than the time taken to perform the indexing/MMP identification step. In our experience of running the algorithm, this is the case with most data sets. However, one would expect the converse to be true for very large or highly connected data sets.

In the implementation of the MMP identification algorithm used for the MLSMR set, the index was written to computer memory (RAM). The amount of memory required for the MLSMR set was 2430 MB. Although this amount of memory is now generally available on a typical workstation, the memory requirements for larger data sets (~1 million) are likely to be prohibitive. However, the index can be written to a file or a database system, which means it can be applied on very large data sets. As a proof of concept study within GSK, a 2 million compound subset of the corporate collection was fragmented, and its corresponding index was written to

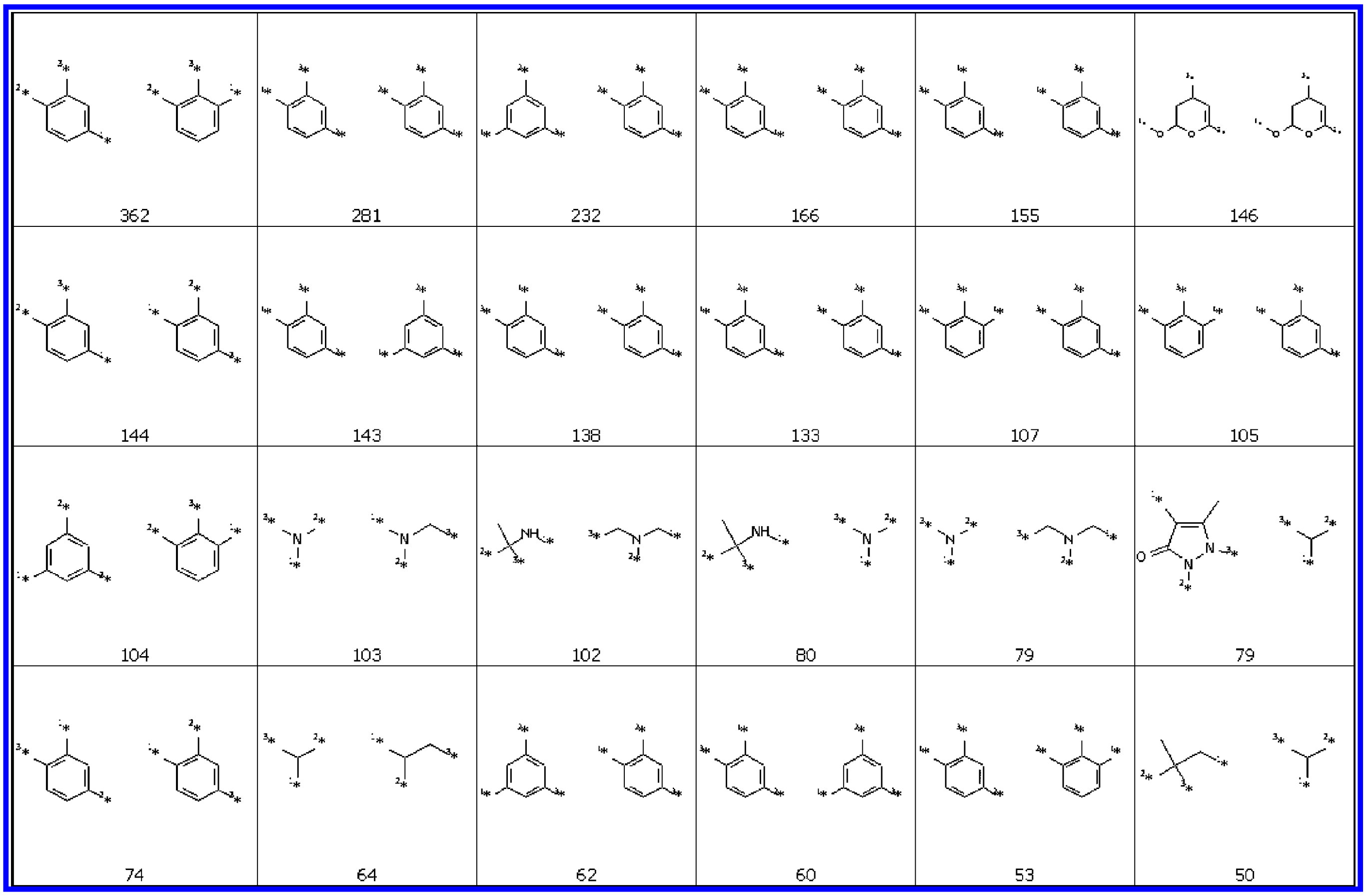


Figure 9. Most frequently occurring core transformations (found from the triple cut fragmentations) in the MLSMR data set. The number of MMPs found for each transformation is also shown.

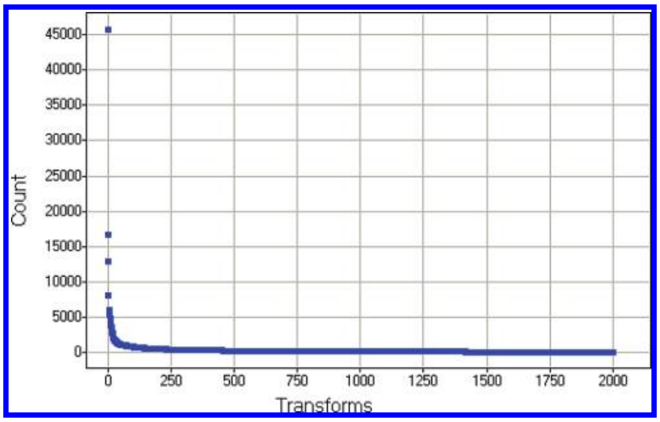


Figure 10. Number of MMPs for the 2000 most frequently occurring transformations found in MLSMR set.

a MySQL[26] database schema. Using this database, it is possible to find all the MMPs of an input compound in the 2 million compound set in around 10 s (note that these timings refer to input compounds that are not part of the 2 million set; to determine the MMPs for a given compound in the database is negligible[27]). With the index written to a database system it is possible to identify the vast majority of MMPs in very large data sets on even modest computational hardware.

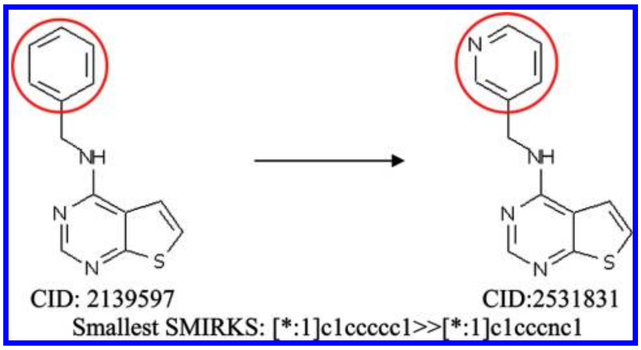


Figure 11. Example of an MMP where the molecular transformation involves a change in a ring atom. The smallest SMIRKS generated by the algorithm is shown. Note, the smallest SMIRKS generated by the algorithm does not correspond to the minimum substructure change for this MMP.

Table 1. Number of MMPs Identified by the Algorithm When Using Different Fragmentation Protocols[a]

fragmentation	number of unique MMPs identified	percentage of total MMPs found
single cuts	4 258 266	80.2%
single and double cuts	5 291 580	99.6%
single, double, and triple cuts	5 310 964	100%

[a] Note the "Percentage of total MMPs found" refers to the proportion of MMPs found using single, double, and triple cut fragmentations in the algorithm.

CONCLUSION

The huge amount of data generated in drug discovery organizations means it is difficult for medicinal chemists to capture all the knowledge available within it. A promising methodology that can be used to capture this knowledge is the MMP methodology. It can be used to systematically data mine a chemical data set to identify interesting structure activity relationships. However, the

MMP methodology requires an appropriate MMP identification algorithm that is capable of finding all MMPs in a large data set, before the potential of the MMP methodology can be fully utilized.

In this paper, we report on a MMP identification algorithm that can be used to identify all the MMPs in large data sets. An implementation of the algorithm was successfully applied to the ~300k MLSMR compound set and ~5.3 million MMPs were identified covering ~2.6 million transformations. The computational efficiency of the algorithm means it can be used to identify MMPs in very large compound sets on modest hardware.

With the reported MMP identification algorithm, it is now possible to systematically mine large chemical data sets. The vast number of MMPs and corresponding molecular transforms found by the algorithm, can be used in conjunction with experimental assay values (and appropriate statistics) to identify transformations that consistently lead to an improvement (and hence identify novel structure–activity relationships). These novel structure–activity relationships are likely to be extremely useful for medicinal chemistry projects as they can be used to help suggest specific structural changes to make to a compound series to improve its properties.

ACKNOWLEDGMENT

The authors thank Stefan Senger, Gianpaolo Bravi, Andrew Leach, Gavin Harper, George Papadatos, and Darren Green for most helpful discussions.

REFERENCES AND NOTES

(1) Topliss, J. G. Utilization of Operational Schemes for Analog Synthesis in Drug Design. *J. Med. Chem.* **1972**, *15* (10), 1006–1011.

(2) Haubertin, D. Y.; Bruneau, P. A Database of Historically-Observed Chemical Replacements. *J. Chem. Inf. Model.* **2007**, *47*, 1294–1302.

(3) Raymond, J. W.; Watson, I. A.; Mahoui, A. Rationalizing Lead Optimization by Associating Quantitative Relevance with Molecular Structure Modification. *J. Chem. Inf. Model.* **2009**, *49* (8), 1952–1962.

(4) Kenny, P. W.; Sadowski, J. Structure Modification in Chemical Databases. In *Chemoinformatics in Drug Discovery*; Oprea, T. I., Ed.; Wiley-VCH: Wienheim, Germany, 2004; pp 271–285.

(5) Sheridan, R. P. The Most Common Chemical Replacements in Drug-Like Compounds. *J. Chem. Inf. Comput. Sci.* **2002**, *42*, 103–108.

(6) Southall, N. T.; Ajay. Kinase Patent Space Visualization Using Chemical Replacements. *J. Med. Chem.* **2006**, *49* (6), 2103–2109.

(7) Gleeson, P.; Bravi, G.; Modi, S.; Lowe, D. ADMET rules of Thumb II: A comparison of the effects of common substituents on a range of ADMET parameters. *Bioorg. Med. Chem. Lett.* **2009**, *17* (16), 5906–5919.

(8) Leach, A. G.; Jones, H. D.; Cosgrove, D. A.; Kenny, P. W.; Ruston, L.; MacFaul, P.; Wood, M.; Colclough, N.; Law, L. Matched Molecular Pairs as a Guide in the Optimization of Pharmaceutical Properties; A Study of Aqueous Solubility, Plasma Protein Binding and Oral Exposure. *J. Med. Chem.* **2006**, *49*, 6672–6682.

(9) Sheridan, R. P.; Hunt, P.; Culberson, J. C. Molecular Transformations as a Way of Finding and Exploiting Consistent Local QSAR. *J. Chem. Inf. Model.* **2006**, *46* (1), 180–192.

(10) Hajduk, P. J.; Sauer, D. R. Statistical Analysis of the Effects of Common Chemical Substituents on Ligand Potency. *J. Med. Chem.* **2008**, *51* (3), 553–564.

(11) Birch, A. M.; Kenny, P. W.; Simpson, I.; Whittamore, P. R. O. Matched Molecular Pair Analysis of Activity and Properties of Glycogen Phosphorylase Inhibitors. *Bioorg. Med. Chem. Lett.* **2009**, *19*, 850–853.

(12) Lewis, M. L.; Cucurull-Sanchez, L. Structural Pairwise Comparisons of HLM Stability of Phenyl Derivatives: Introduction of the Pfizer Metabolism Index (PMI) and Metabolism-Lipophilicity Efficiency (MLE). *J. Comput.-Aided Mol. Des.* **2009**, *23*, 97–103.

(13) SMARTS, SMIRKS, findsub. Daylight Chemical Information Systems, Inc., Aliso Viejo, CA, http://www.daylight.com/ (accessed October 2009).

(14) Lewell, X. Q.; Judd, D. B.; Watson, S. P.; Hann, M. M. RECAP–Retrosynthetic Combinatorial Analysis Procedure: A Powerful New Technique for Identifying Privileged Molecular Fragments with Useful Applications in Combinatorial Chemistry. *J. Chem. Inf. Model.* **1998**, *38*, 511–522.

(15) Perl programming language. http://www.perl.org/ (accessed October 2009).

(16) Weininger, D.; Weininger, A.; Weininger, J. L. SMILES. 2. Algorithm for Generation of Unique SMILES Notation. *J. Chem. Inf. Comput. Sci.* **1989**, *29*, 97–101.

(17) SMARTS used: “[*]!@!=[*]”.

(18) NIH Molecular Libraries Small Molecule Repository. http://mlsmr.glpg.com/ (accessed October 2009).

(19) PubChem. http://pubchem.ncbi.nlm.nih.gov/ (accessed October 2009). The search term: “MLSMR” was used to return the NIH Molecular Libraries Small Molecule Repository Set.

(20) Compounds were removed if they were found to be mixtures after an in-house desalting procedure was applied (56 compounds) or if they contain more than 100 non-hydrogen atoms such as peptides (7 compounds).

(21) Compounds were removed if they contained nonstandard isotopes (8 compounds) or if they did not have a chemically tractable bond [C-(N,O,S)] (42 compounds) or contained nonorganic elements (that is, not in the set (C, N, O, P, S, halogen, B, or Si) (46 compounds).

(22) The calculation was performed on a single core of a dual core Intel (R) Xeon(TM) 3.00 GHz cpu (2048 Kb cache size).

(23) Zipf’s Law. http://en.wikipedia.org/wiki/Zipf%27s_law (accessed October 2009).

(24) SMARTS changed from “[*]!@!=[*]” to “[#6+0;!$(*=,#[!#6])]!@!=!#[*]”.

(25) It was noted by one of the reviewers that a maximum size parameter of 10 non-hydrogen atoms will miss some (perhaps key) transformations (e.g., C1(C2=CC=CC=C2)=CC=CC=C1 (biphenyl) to C1(C2=CC=CC=C2C3)=C3C=CC=C1 (fluorene)). Therefore, an appropriate value for the maximum size parameter needs to be chosen. This depends on the size of the dataset and the size of the molecular transformations one would like to find.

(26) MySQL, Sun Microsystems. http://www.mysql.com/ (accessed October 2009).

(27) Determined by calculating the average time to retrieve the MMPs for 20 randomly selected compounds from the database. The BENCHMARK function available within MySQL was used to determine the time it took to perform the SQL query (a thousand times) to find the MMPs (for each of the 20 twenty compounds). The average time to run a thousand SQL queries was 0.88 seconds which equates to an average time of 0.00088 seconds for a single SQL query to retrieve the MMPs for a database compound.

CI900450M