JOURNAL OF CHEMICAL INFORMATION AND MODELING

Article

pubs.acs.org/jcim

A System for Encoding and Searching Markush Structures

David A. Cosgrove,*[,†] Keith M. Green,[‡] Andrew G. Leach,[§] Andrew Poirrette,[‡] and Jon Winter[∥]

[†]Discovery Sciences Computational Sciences, AstraZeneca, Alderley Park, Macclesfield, Cheshire SK10 4TG, United Kingdom

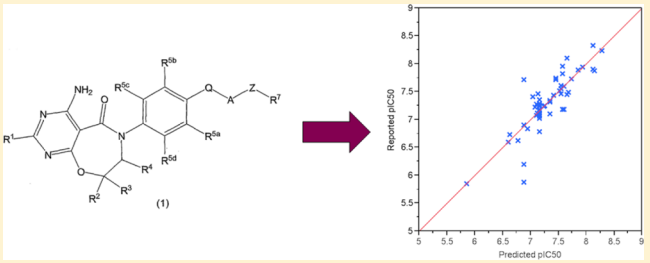
[‡]Research and Development Informatics, AstraZeneca, Alderley Park, Macclesfield, Cheshire SK10 4TG, United Kingdom

[§]CVGI iMED, AstraZeneca, Alderley Park, Macclesfield, Cheshire SK10 4TG, United Kingdom

[∥]Oncology iMED, AstraZeneca, Alderley Park, Macclesfield, Cheshire SK10 4TG, United Kingdom

S *Supporting Information*

ABSTRACT: The encoding and searching of generic chemical structures, so-called Markush structures, have received little attention in the literature of late. The ability to encode and search these complex entities is of use in various branches of chemoinformatics. We describe a general language for encoding Markush structures and algorithms for searching them and give three examples of the utility of such a system: development of general Free–Wilson analyses of chemical series, detection of controlled substances within a large database of molecular structures, and searching of large databases of virtual compounds.



INTRODUCTION

A Markush structure is a generalized chemical formula that compactly describes a set of closely related chemical structures. It is named after Eugene Markush who in 1924 was awarded a patent for pyrazolone dyes[1] described using a generic description of the starting materials. Markush structures are commonly used in two areas in chemoinformatics: in the description of chemical libraries and in chemical patents such as those used to protect the invention of new compounds. The two uses are subtly different in that in the former it is normal for the variable parts to be exactly specified so that a full enumeration of the library is in principle possible, computer time and disk space permitting. In the latter use, the variable parts are normally more loosely described (phrases such as "alkyl" and "heteroaromatic ring" are common) such that it is not generally feasible to enumerate unambiguously all molecules encompassed by the Markush structure. A particular difficulty posed by Markush structures in chemical patents is that they frequently contain differing definitions for common chemical terms as required by the structures in question. One patent might define alkyl as containing between one and six carbon atoms, for example, where another might set the upper bound as eight. The language and associated software described herein can deal with both such types of Markush structure.

With one notable exception, the searching of Markush structures from chemical patents has received little attention in the cheminformatics literature. A group of researchers at Sheffield University produced a series of 23 papers between 1981 and 1996 that described in great detail their analysis of the problem and the system they created to solve it. Helpfully, the 23rd paper[2] in the series is a summary of the previous 22 and contains references to them all.

The Sheffield work formed the basis or heavily influenced the development of the two commercially available patent searching systems, MARPAT[3] (a service from CAS[4]) and the Merged Markush Service (MMS)[5] of Thomson-Reuters.[6] For many years, these two systems were only available for searching on a for-fee basis as an online service tended to be used only by trained specialists. Recently, there has been a resurgence of interest in the subject. ChemAxon[7] have announced the interfacing of their JChem software to the MMS databases, and SciFinder from CAS now has limited structure searching of the MARPAT database. Digital Chemistry[8] have discussed extending their Torus Chemical Library database software to allow searching of more general Markush structures.[9] Deng et al.[10] have described a system, MarVis, that uses extensions to the SMILES language and bespoke graphical software to assist in the interpretation of results from searches of MMS. Downs and Barnard[11] have recently and comprehensively reviewed the state of Markush searching systems. ChemAxon have also extended their Marvin molecular sketcher and JChem chemical database system and continue to develop it to allow the encoding and searching of Markush structures.[12] This was not available when the work developing the Periscope system described herein commenced. Finally, InfoChem have announced the ChemProspector[13] project, funded by the German governement, that aims to build software to extract automatically the core and R Groups from chemical patents and store them in a searchable database.

There are a number of reasons why it is desirable to be able to describe and search Markush structures outside the

Received: January 20, 2012
Published: July 17, 2012

dx.doi.org/10.1021/ci3000387 | *J. Chem. Inf. Model.* 2012, 52, 1936–1947

commercial indexed systems mentioned above. This paper exemplifies three such uses that underlie the motivation for the creation of the Periscope system: Free−Wilson-style QSAR analyses of a homologous series of compounds such as those claimed in a chemical patent, monitoring the presence of controlled substances within compound databases, and searching a very large library of virtual chemical libraries on the order of 10^{12} compounds.

The Periscope system has three major pieces: an XML-based language for describing Markush structures (the Markush Input Language, or MIL), a graphical program (Menguin) for encoding Markush structures in the MIL format, and a search and analysis program (i3am) for investigating whether one or more specific chemical structures are described by the Markush structure.

METHODS

The Markush Input Language. The Markush structure is defined by way of XML (eXtensible Markup Language). XML is a set of rules for encoding data or documents in machine-readable form. It does not describe a language itself; it merely sets a format and style in which a system designer is free to develop a language. It is particularly useful for data files where there is varying content for different records because each data record is given a unique "tag" or label to which is usually attached associated data. There are a number of application programming interfaces (APIs) that allow for the easy reading, writing, and parsing of XML format files. The text-based nature of the files means that they are easy to transfer between different computers and operating systems, in a way that, for example, binary files are not. On the negative side, they can be lengthy and verbose, sacrificing compactness for ease of use and convenience. XML is used in the Chemical Markup Language (CML),[14] and there are elements of the Polymer Markup Language (PML)[15] derived from CML that its creators suggest might be used for representing Markush structures. We do not believe that this power has been exploited in practice.

The XML format lends itself very well to the specification of Markush structures for a number of reasons, not least being the fact that very few of the properties of the R Groups in the Markush are compulsory and so can be left out of the XML if not needed, in contrast to a fixed-format tabular style, where every property would need an entry. Also, the XML format allows for a tree structure to be built (tagged entries or nodes can have nested subentries), which is also useful for specifying substituents to R Groups.

Markush structures are conveniently considered as a "core" substructure substituted by variable groups (the R Groups). R Groups can themselves be substituted, so that a recursive multibranched tree is described, and the R Groups may be attached to the core at fixed or variable positions. Variation is also frequently specified in the core itself, so it is possible to treat it as a special case of an R Group that forms the root of the R Group tree. Each R Group has a unique name (e.g., R1, W), but there can be multiple elements with the same name (e.g., W is alkyl, halogen, etc.). In principle, the tree structure of the Markush could be mirrored by a corresponding structure in the XML. However, we found it simpler to have a relatively shallow tree with the full depth of the Markush structure being built up as the XML is parsed.

We define two main types of R Groups: Exact and Inexact. An Exact R Group is one for which a precise substructure may be specified, in our case either by SMARTS[16] string or Accelrys[17] RGFile.[18] Inexact R Groups are those for which no substructure can be defined but which nevertheless can be characterized precisely by specifying, for example, element and bond counts, ring membership, etc. We also have a third, rarely used, type: Fused. This is for convenience and allows for cases where a cyclic system is defined in terms of fused subtypes. Normally, for example, one will have defined heteroaromatic and aliphatic ring systems, and the Fused R Group allows a larger ring system to be specified in terms of the subrings, for example, an aliphatic carbocycle fused to a heteroaromatic ring.

Both SMARTS strings and RGFiles are expressive languages allowing the specification of complex substructural queries. The MIL allows full use of these features in the Exact R Group definitions. It might be convenient, for example, to define a halogen substituent as a single SMARTS string [F,Cl,Br,I] rather than as four separate entries, one for each element. Equally, one could use a complex SMARTS definition to allow the definition of a six-membered aromatic ring, with no ring fusions, where some of the atoms might be nitrogen or carbon. This can save time in the Free−Wilson analyses (see below), when SMILES strings of the substituent portions are used as the descriptors as it avoids the need to define an exact R Group for all possibilities present in the molecule set. The user must, however, be aware of limitations that the matching software imposes when using this flexibility. One can combine several different SMARTS definitions into a larger definition using boolean logic. The SMARTS string [$(a1aaaaa1),$(a1aaaa1)], for example, defines an atom in a five- or six-membered aromatic ring. Matching it to a benzene ring will produce six hits, each of one atom only, and to furan will give five one-atom hits. This will almost certainly not produce the search result that the user expects or wants. To specify a five- or six-membered ring in its entirety, it is necessary to use two separate R Group definitions, one with the six-atom SMARTS and the other with the five-atom one. As described below, the searching algorithm uses the substructure definition to remove from the query structure all possible sets of matching atoms, one set at a time. As such, it is neutral as to the complexity of or means of specifying (SMARTS or RGFile) the substructure.

Each R Group has a number of properties that together define which substituents it matches. Some are common to all three R Group types, while others are specific to a particular type. Brief descriptions are given here; full details are in the Supporting Information, along with several example files. A simple example demonstrating some of the more commonly used elements of the language is shown in Figure 1 that encodes the Markush structure depicted in Figure 2.

Examples of compounds that fall within and outside the Markush are shown in Figures 3 and 4, respectively.

As a further example, we have attempted to encode the structure in Markush's original patent.[1] Unfortunately, it contains a number of ambiguities and unclear definitions, so a certain amount of interpretation was required. For example:

- He speaks of "aniline or its homologues (such as toluidine, xylidine, etc.)", which we have taken to mean aniline optionally substituted by one or more methyl groups.
- Some functional groups appear in his list of example reagents but not in his list of claims. For example, he speaks of "halogen substituted pyrazolones (such as dichlorsulphophenylmethyl pyrazolone)." We have included these.

dx.doi.org/10.1021/ci3000387 | J. Chem. Inf. Model. 2012, 52, 1936−1947

```
<?xml version="1.0"?>
<markush>
  <!-- Core -->
  <r_group>
    <name>Core</name>
    <type>Exact</type>
    <text>Benzene Core</text>
    <smarts>c1ccccc1</smarts>
    <substituent>
      <subst_name>R1</subst_name>
      <subst_attach_point>1</subst_attach_point>
    </substituent>
    <substituent>
      <!-- R2 can be ortho, meta or para to R1, and there can be
           1 or 2 of them -->
      <subst_name>R2</subst_name>
      <subst_min_count>1</subst_min_count>
      <subst_max_count>2</subst_max_count>
      <subst_attach_point>2 3 4</subst_attach_point>
    </substituent>
  </r_group>
  <!-- R1 Fluoro or Methyl-->
  <r_group>
    <name>R1</name>
    <type>Exact</type>
    <text>R1_F</text>
    <attach_by>1</attach_by>
    <smarts>F</smarts>
  </r_group>
  <r_group>
    <name>R1</name>
    <type>Exact</type>
    <text>R1_Methyl</text>
    <attach_by>1</attach_by>
    <smarts>[CH3]</smarts>
  </r_group>
  <!-- R2 C1-C6 alkyl or alkenyl, C3-C6 cycloalkyl -->
  <r_group>
    <name>R2</name>
    <type>Inexact</type>
    <text>Alkyl or alkenyl</text>
    <branched>dont_care</branched>
    <cyclic>false</cyclic>
    <aliphatic>true</aliphatic>
    <element_count>
      <type>C</type>
      <min>1</min>
      <max>6</max>
    </element_count>
    <bond_count>
      <type>2</type>
      <min>0</min>
    </bond_count>
  </r_group>
  <r_group>
    <name>R2</name>
    <type>Inexact</type>
    <text>CycloAlkyl</text>
    <branched>true</branched>
    <cyclic>true</cyclic>
    <aliphatic>true</aliphatic>
    <element_count>
      <type>C</type>
      <min>3</min>
      <max>6</max>
    </element_count>
  </r_group>
</markush>
```

Figure 1. Simple MIL file exemplifying some of the more commonly used constructs in the language. The core is a benzene ring, substituted by one R1 group, which can be methyl or fluoro, and one or two R2 groups, *ortho*, *meta* or *para* to R1. R2 can be C_1-C_6 alkyl or alkenyl, or C_3-C_6 cycloalkyl.

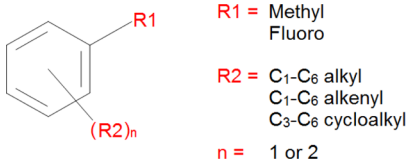


Figure 2. Markush structure encoded in Figure 1.

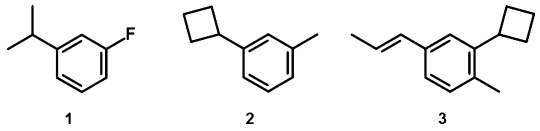


Figure 3. Examples of compounds that fall within the example MIL shown in Figure 1.

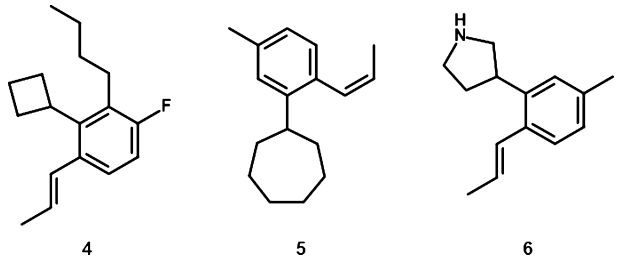


Figure 4. Examples of compounds that fall outside the example MIL shown in Figure 1. Molecule **4** has too many R2 groups. In molecule **5**, the ring is too large; and in molecule **6**, R2 contains a heteroatom.

- The pyrazolone groups claimed can exist in more than one tautomer, so in these cases we have included both the keto and enol possibilities.

Encoding the Markush into the MIL using Menguin took about two hours, of which perhaps thirty minutes was the actual encoding; the rest was taken up by searching various references to understand the chemistry and hence the claims. The full MIL file is included in the Supporting Information. We have used it to search a snapshot of PubChem[19] from November 2011 containing slightly over 27 million unique SMILES strings. There were seven hits, which are displayed in Figure 5. The search took approximately seven hours on a single processor.

It was only during the encoding process, with the necessity of translating the words in it into substructures and specific counts of atoms and bonds, that the full extent of the incomplete definitions in the patent became apparent. One potential use of the Periscope system might therefore be the encoding of the Markush structures in patents under preparation, prior to their filing. This could assist in the removal of possible confusion because during the encoding, each R Group definition is converted into an unambiguous rule; if the text cannot be so converted, it is clear that the text is not precise and might then be reworded. One could also query the encoded Markush with the example structures from the proposed patent, so helping to ensure that they are all included within the encoded Markush structure. The result would hopefully be a more tightly worded patent with fewer inconsistencies. A patent is no different from any other legal document in that those that are more loosely or ambiguously worded are more likely to result in litigation. Such litigation is frequently very expensive, so the Periscope system might be a useful cost-containment tool.

General Properties Common to All R Group Types.

- **name**: Each R Group has a name (e.g., "R1") by which it is referred to in the MIL file. Multiple R Groups can have the same name.
- **text**: An optional short piece of text (e.g., "R1 Alkyl") that can be used to distinguish one instance of an R Group from another. It is generally the case that each instance of an R Group has a different text tag, so that analyses can distinguish the particular variant of an R Group in a particular molecule.

dx.doi.org/10.1021/ci3000387 | J. Chem. Inf. Model. 2012, 52, 1936−1947

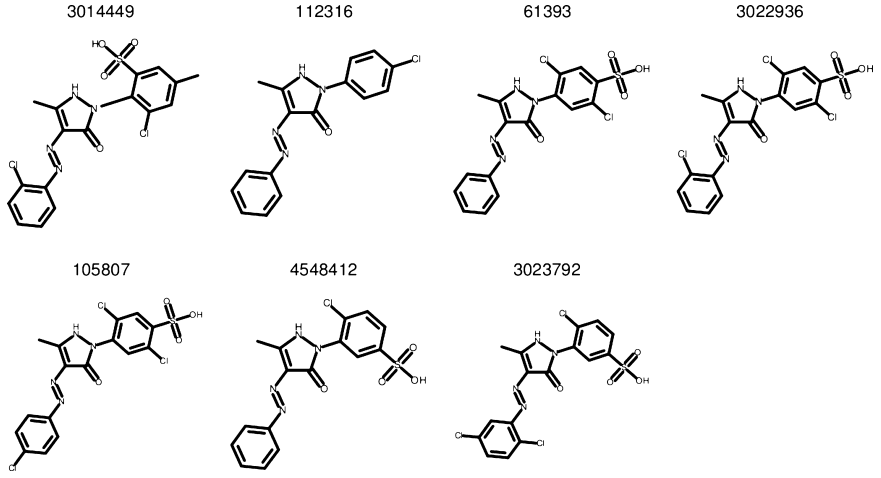


Figure 5. Hits from a search of PubChem using the MIL encoded from Markush' original patent for pyrazolone dyes.

- **remark**: An optional longer text string, e.g. "C1−C6, branched or unbranched." The purpose is to document the R Group to make the file more informative to humans.
- **attach_by**: An optional property that describes how the R Group attaches to the parent moiety (the Core, or an R Group higher in the tree). By default, any atom in the R Group can be attached to the parent, and the *attach_by* property allows finer control.
- **attach_by_bond**: Similar to the *attach_by* property, this property is a number between 1 and 3 and defines the order of the bond by which by which the R Group attaches to its parent. It defaults to 1.
- **chain_max_count, chain_min_count, chain_attach_point**: These are used when the R Group is a repeating unit forming a chain, such as in "$-[C(R_m)(R_n)]_{1-4}-$." The *chain_attach_point* property is used in multi-atom repeating units to specify how the chains are joined together, allowing, for example, the distinction between an *n*-propyl and *i*-propyl repeating unit.
- **substituent**: This property defines a subnode in the MIL and denotes substituents to the R Group. There are a number of properties for the subnodes that control things like how many instances of the substituent are allowed, what the substituent attaches to, etc. These are described in more detail below.

Properties Particular to Exact R Groups. There are three properties possessed exclusively by Exact R Groups and one specialization of a general property:

1. **smarts**: This specifies the substructure of the Exact R Group as a SMARTS string.
2. **molfile**: This specifies the substructure of the Exact R Group using a string corresponding to an Accelrys RGFile. It is normally created by sketching the structure using Accelrys' JDraw[20] applet embedded in the program Menguin.
3. **attach_by**: For Exact R Groups, the *attach_by* property is a string containing zero or more integers in a space-separated list, counting from 1, optionally followed by a "|" and 1 or two SMARTS strings, separated by a space. The numbers in the first part specify the sequence numbers of the atoms in the substructure that can be used to attach the substituent to the parent. So, for example, if the substructure is specified by the SMARTS string "C(=O)NCC", an *attach_by* property of "1" would specify attachment only by the carboxyl carbon. The first part may also contain a '+' or '−' symbol followed by a second set of integers. This is used to denote cases where the R Group may have more than one attachment atom, either because it is bidentate or fused onto the parent. The second, optional, part of the *attach_by* property defines the environment of the atoms in the parent group to which the R Group attaches.

Figure 6 shows examples of exact R Groups, including an example where atoms in the ring can vary. This is achieved using the final property specific to Exact R Groups, *variation*, a complex and rarely used feature, whose description is reserved for the Supporting Information.

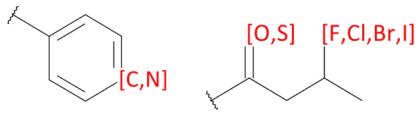


Figure 6. Examples of Exact R Groups. The left-hand group shows in-ring variation at the 4 position. The squiggle marks the point of attachment to the parent moiety.

Properties particular to Inexact R Groups. Inexact R Groups are much more general than Exact Groups, and so they need more properties to define them. There are three properties, *branched*, *cyclic*, and *aliphatic*, that describe general features of the R Group. They can have the values *true*, *false*, or *dont_care*. All combinations of values can be set for the three properties, but clearly some, such as *aliphatic* is *false* (i.e., it is an aromatic group) combined with *cyclic* is *false* (denoting an acyclic group) are not sensible. These inconsistencies are not checked for, but it is somewhat unlikely that such an R Group will match anything in a real molecule.

There are three further properties that form subnodes in the MIL: *element_count*, *bond_count*, and *smarts_count*. They have properties *min*, *max*, and *type* and allow the specification of allowed counts of different elements, bonds, or SMARTS patterns. The type is respectively an atomic symbol, a bond order (1, 2, 3, or 5 for aromatic), or a SMARTS pattern. The *min* and *max* properties give the range of values. If the minimum is unspecified, it is assumed to be 1, and if the maximum is omitted, there is no maximum. A maximum of 0 means the feature is not allowed in the R Group. For example,

dx.doi.org/10.1021/ci3000387 | J. Chem. Inf. Model. 2012, 52, 1936−1947

if the R Group is for a ring, and fused rings are not wanted, one could specify a *smarts_count* of type "[x3,x4]" and a *max* of 0, where "[x3,x4]" indicates an atom with 3 or 4 incident ring bonds. It is also possible to specify how far from the parent moiety the SMARTS pattern can appear in the R Group. For example, if the R Group must contain an amide group but not as the attachment point, one could use the SMARTS pattern "NC=O 0=1_9999 1=1_9999 2=1_9999" with minimum and maximum counts of 1, which denotes that there must be one and only one amide group and the first, second, and third atoms must all be at least 1 bond and not more than 9999 bonds from the parent atom to which the group is attached (atoms are numbered starting from zero in this case). For the *element_count* subnode, as well as atomic symbols, one can use a type of "*" to indicate any atom, or "Het" for noncarbon and non-hydrogen atoms. A combination of elements or bonds can be indicated using the "|" symbol to mean "OR" as in "O|N|S", meaning oxygen or nitrogen or sulfur.

The *attach_by* property for Inexact R Groups requires a list of one or more atomic symbols, giving the elements in the R Group that can be attached to the parent moiety. So for a heterocycle containing C, S, N, and O elements, an *attach_by* string of "N" dictates that the R Group can only attach to the parent via a nitrogen atom. In an analogous way to the Exact R Group, it is also possible to specify fused and bidentate attachment by way of the symbols "|", "+", "−" and SMARTS patterns.

Properties Particular to Fused R Groups. A Fused R Group is a composite of two Exact, Inexact, or Fused R Groups and is used to encode fused ring systems. There are two additional properties over the general ones, *first_r_group* and *second_r_group*, which give the names of the two constituent R Groups, which are required to be cyclic. The *attach_by* property in this case is used to specify whether attachment to the parent is via the first R Group or the second and if not specified defaults to either. Thus, if the Fused R Group is a combination of an aromatic ring and an aliphatic one, it is possible to state whether the former or latter is attached to the parent.

In principle, the Fused R Group type is redundant because it is normally possible to define a fused ring system precisely as an Inexact R Group. However, consider the case where one has two different ring R Groups. One is an aliphatic heterocycle coming in three size ranges, each of which has different numbers of unsaturated bonds and heteroatoms in each size. For example, it might be defined as a ring of five to eight atoms, with the following properties. When the ring has five atoms, one of them can be oxygen, nitrogen, or sulfur, and there may one unsaturated bond. When the ring has six or seven atoms, there can be one or two oxygen, nitrogen, or sulfur atoms and up to two unsaturated bonds, and when the ring has eight atoms, there can be up to three oxygen, nitrogen, or sulfur atoms and up to three unsaturated bonds. This would be encoded using three different R Group definitions in the MIL. The other ring R Group might be an aliphatic carbocycle also in three size ranges, and again with different numbers of unsaturated bonds in the different sizes. This, too, would be encoded with three different R Group definitions in the MIL. One might want to define a new R Group that is the heterocycle fused to the carbocycle. With the three R Group definitions for each in the MIL, this would give rise to nine different combinations, which in turn encode a substantial number of fused rings. It might be possible to encode these nine combinations as separate Inexact R Groups, but it would be complicated. Specifying that heteroatoms could be in only one of the constituent rings would not be trivial. The Fused R Group makes it very straightforward to specify in the MIL file. This convenience comes at the expense of transferring the complications to the searching program i3am, but at least the user is shielded from these difficulties.

More details of the Fused R Group, including subtleties when searching with them, are given in the Supporting Information.

Properties of Substituent Subnodes. All R Groups can contain substituents, specified by elements in the R Group definition of type *substituent*. These can contain the following subelements:

- **subst_name**: This is compulsory and gives the name of the substituent. The substituent must be defined elsewhere in the MIL, but it does not matter whether this is before or after it is used by an R Group.
- **type**: The type of a substituent can be either *plain* (the default, and far and away the most common type), *chain*, or *global*. A *chain* substituent is one only found on a repeating R Group, so that in the R Group $[-C(R_m)-(R_n)-]_{1-4}-X$, the two substituents R_m and R_n are *chain* substituents, and the X is a *plain* substituent that can only be attached to the last member of the chain. A global substituent is one that can be on the parent R Group and any of its dependent substituents, allowing for situations such as "phenyl substituted by alkyl, either of which is further substituted by one or more halogen atoms."
- **subst_min_count, subst_max_count**: Gives the minimum and maximum number of substituents of this type that can be attached to the R Group. If neither of these is specified, they both default to 1. If either is specified, the other defaults to 1 and no maximum, respectively. A minimum count of 0 implies that the group is optional.
- **subst_attach_point**: The exact specification of this is different for Exact and Inexact R Groups, but in both cases, it defines where on the R Group the substituent may be found. In the former case, it is a 1 or more numbers (counting from 1), being the sequence numbers of atoms in the SMARTS string (reading left to right) or RGFile (atom record order). For Inexact R Groups, it is 1 or more atomic symbols, giving the element types that might bear the substituent.

Encoding a Markush structure. In principle, a Markush structure can be encoded by typing the MIL directly with a simple text editor, and indeed, that is how it was done originally. However, this is tedious, error-prone, and requires a detailed knowledge of both XML and the MIL. Instead, we have written the program Menguin (*M*arkush *En*coding *G*raphical *U*ser *In*terface) to assist and make Markush structure encoding and searching practical for a much wider audience. Menguin is a Rich Internet Application (RIA) running in a browser using Adobe Flex[21] in the client together with a Java back-end. Exact R Groups are created using Accelrys' JDraw Java applet, and Inexact R Groups are defined using a dialogue sheet. There is a large and growing dictionary of preprepared definitions that can be incorporated with modifications if necessary. Some Exact R Groups may also be specified by name, using OpenEye's Lexichem[22] tool to perform the name to structure translation. A screenshot of Menguin is shown in

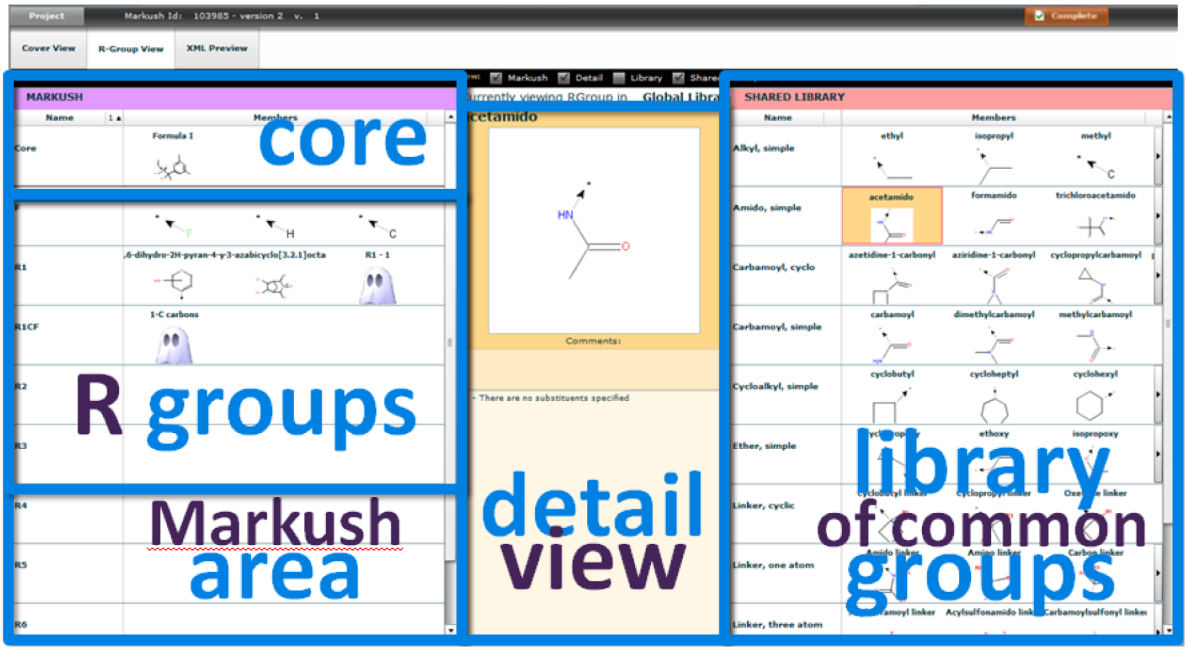


Figure 7. Screen shot of the Menguin application for defining Markush structures.

Figure 7. A typical Markush structure can be encoded in a few hours using Menguin.

Searching with the Markush Structure. Having encoded a Markush structure, one is then able to use the program i3am (*is it in a M*arkush) to establish whether one or more structures from a query file are described by the Markush. At its most basic, this results in a simple yes or no. However, it is also possible to return information about which parts of each query structure match which parts of the Markush structure. If the query structure is not contained in the Markush, i3am will identify those parts which are, those that are not or, if appropriate, where the query molecule lacks something that the Markush requires. The program i3am can also produce for each query molecule a table of which example of each R Group is found in each molecule, using the text property of the R Group or a SMILES string of the matching part of the molecule. In the latter case, a xenon atom is used to mark where the fragment attached to its parent moiety. The program i3am is implemented in C++ using OpenEye's OEChem toolkits[23] for molecule file parsing, substructure matching, and general molecular manipulation.

The Sheffield system used a sophisticated and complicated series of fragment screening procedures to make the searching of Markush systems feasible in a reasonable length of time. However, improvements in computer speed, memory size, and availability and also in cheminformatics toolkits in the intervening 20 years mean that the problem is now amenable to a rather cruder brute-force approach. As a general rule, there are two ways to increase throughput speed for a computational problem: one can optimize the algorithm or apply more computer resources to it. We have opted for the latter, which is less elegant but quicker to program. The search performed by i3am is also much simpler than those possible using the Sheffield system and its commercial derivatives. Here, we concentrate on searching single molecular structures against the Markush, as opposed to the more sophisticated searches the others offer where complex substructure queries are made against databases of hundreds of thousands of Markush definitions.

The search algorithm in i3am is a recursive, depth-first procedure that attempts to maximize the number of atoms in the query structure that match something in the Markush. If there were only one possible definition for each R Group, it would be a straightforward tree-search, but matters are somewhat more complicated because at each point in the search there are normally multiple options that must be explored and scored, particularly in the case of Inexact R Groups. The search is further complicated by the fact that a single R Group definition might match more than one set of atoms in the remaining structure, and all these possibilities must also be accounted for. There is thus a considerable combinatorial problem. The algorithm adopted is a greedy one working from the leaves of the tree back to the core where the best-scoring outer submatches are transferred back up the tree at each step backward in the recursion. In principle, this could result in a suboptimal match, but we have not seen this in practice. For some structures a search can take several minutes, and the more general the Markush and the larger the query structure, the longer the search time. In practice, i3am achieves less than a second per query molecule for a typical Markush structure. At present, the Core group, which all Markush structures are defined from, must be one or more Exact R Groups. Frequently in chemical patents, multiple core substructures are given, either explicitly or by use of, for example, variable ring sizes or optional element types for atoms. The MIL allows for this, and the search in i3am proceeds once for each such core, keeping the best-scoring match for all cores. The restriction that the core must by an Exact R Group is for efficiency and practicality, as it gives a clear starting place for the search. In principle, an Inexact R Group could be used, but run times would be expected to be much longer. If the Core is not found in the molecule, the search terminates immediately.

The algorithm proceeds as follows:

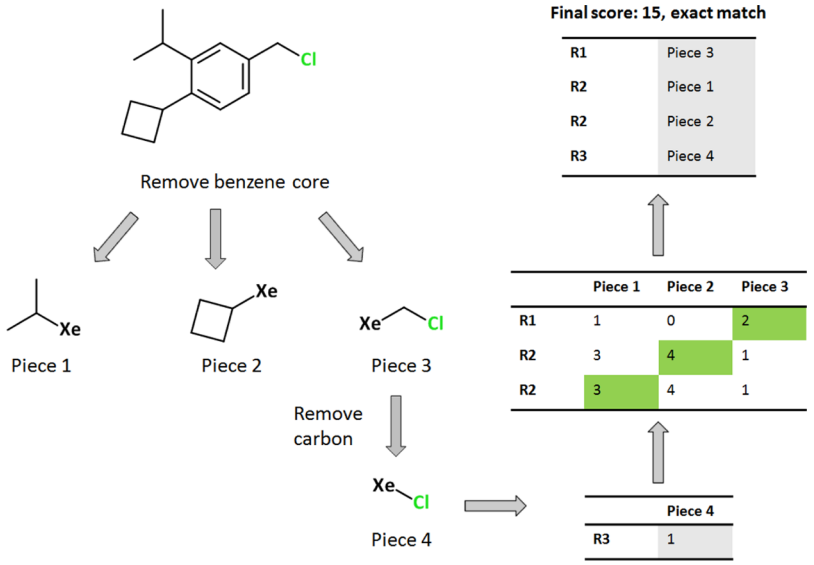


Figure 8. Diagram showing the progress of the algorithm for matching a molecule against a MIL. The Core (phenyl) is matched against the molecule, and the matching atoms are removed, creating 3 pieces. The xenon atom marks the point of attachment of the substituent to the core. Each substituent piece is matched against the definitions for R1 and R2, and the number of atoms matching tabulated. R1 is further substituted by R3, so the process is repeated recursively. The LAP algorithm is used to select the highest scoring match combination at each level in the search tree, with the best score being transferred to the table at the higher level.

1. Set the list of R Groups currently being considered to those named "Core" and the molecule remaining to be considered to the whole molecule.
2. For the next R Group in the current list, match the R Group to one or more atoms in the remaining molecule, taking account of the attachment point of the molecule portion. For each match in turn, remove the matching atoms from the remainder, possibly giving one or more portions that must be matched to any substituents that the R Group contains.
3. If the current R Group has no substituents, return the number of atoms in the matching piece as the score for that R Group.
4. If the current R Group has substituents, recursively match each substituent against each remaining molecular portion forming a rectangular matrix of scores for each substituent against each molecular portion, where the score is the number of atoms that match the R Group. There may be more substituents than molecular portions or vice versa, meaning that the query is not an exact match, but proceed anyway to find the best-scoring partial match.
5. Use the Linear Assignment Problem algorithm of Carpaneto et al.[24] to find the best scoring combination of matches of substituent R Group to remaining molecular portion. Add this score to the match score for this R Group.
6. If the total match score is the best so far, note this.
7. If all atoms in the current remaining portion are matched, consider this the final answer and break out of loop.
8. If there are still R Groups in the current list, go to 2.

Figure 8 shows the operation of the algorithm for a MIL, which is an extension of the one shown in Figure 1. In this example, R1 is also allowed to be a methyl group substituted by a halogen, as shown in Figure 9.

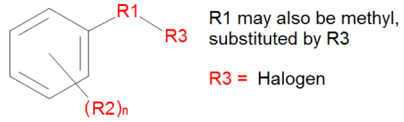


Figure 9. Markush structure used for example of i3am search algorithm progression depicted in Figure 8. R1, R2, and *n* as defined in Figure 1.

It should be noted that there may be more than one way of mapping the Markush structure onto a query molecule. In this case, because of step 7 above, only the first such match will be reported and that will depend on the order of the R Group definitions in the MIL file. In principle, the search could explore all possibilities and return multiple matches, but this is not done for reasons of efficiency.

Matching an R Group to a Molecular Portion. In step 2 of the algorithm above, the current R Group is matched to the remaining molecule portion, the matching atoms removed, and any atoms remaining kept to pass into step 4. The exact procedure is different for the three different R Group types.

Exact R Groups. This is relatively straightforward and is achieved using the OESubSearch class of the OEChem toolkit, which can use both SMARTS and Accelrys RGFiles as input. Allowance must be made for the possibility of multiple matches. Only those matches that include the point of attachment to the parent need be kept, but there are still possibilities for multiple matches, most commonly with ring systems where for reasons of symmetry multiple matches involving the same atoms in different orders are common. All appropriate matches are used in turn as a basis for the next step in the search and score process.

Inexact R Groups. These are somewhat more complicated and time-consuming to deal with than Exact R Groups, owing to the less precise nature of the R Group description. Starting at the attachment point, all contiguous subportions of the molecule are created out to the end of the molecule, not including atoms in the parent moiety. If adding an atom to the

dx.doi.org/10.1021/ci3000387 | J. Chem. Inf. Model. 2012, 52, 1936−1947

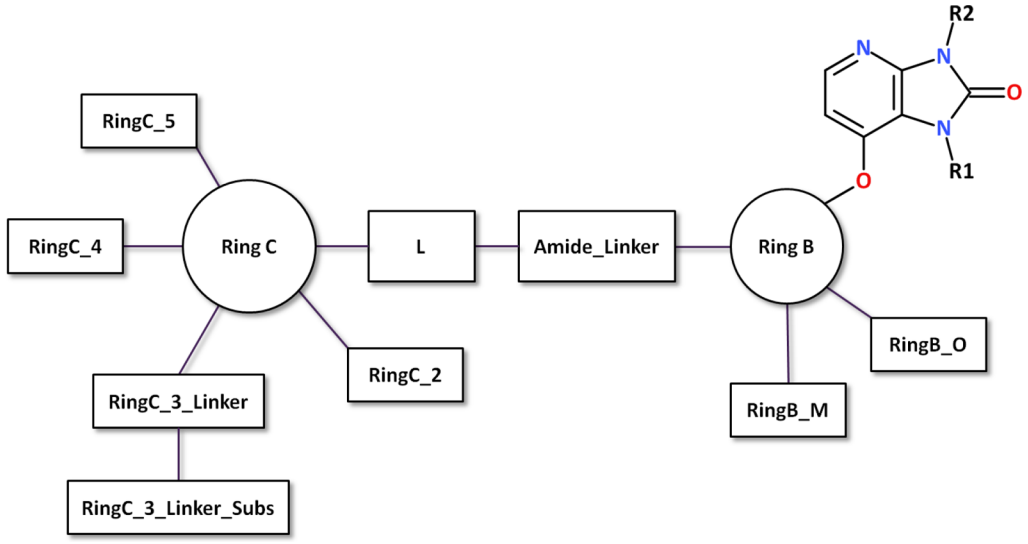


Figure 10. Markush diagram of the structures used in Free–Wilson-style analysis of structures from Yang et al.[27].

current subportion would cause it to exceed some parameter of the R Group, such as if the atom was aromatic and the R Group is defined as aliphatic or it is a second nitrogen atom and the R Group can only contain one then that branch of the search tree can be pruned off. Nevertheless, the number of molecule pieces that match the R Group definition and which must be passed into step 2 of the algorithm above is frequently large, with significant consequences for the time for the whole search, particularly if the R Group is also substituted by further Inexact R Groups.

Fused R Groups. These are the most complicated of the groups to deal with. Either of the two subgroups may be Exact or Inexact and, unless explicitly stated in the MIL, both of them must be considered as the group that attaches to the parent moiety with the other being considered in the second phase. They must both be cyclic structures. The search is complicated because the two groups must by definition share at least one atom, the point of fusion between the two rings. The normal search procedure is carried out using the first R Group, and the atoms for all matches removed from the molecule, except those that have 3 or 4 ring bonds incident upon them. These atoms could be the point of fusion, so must be present for the next stage, where the second R Group is matched against the remaining atoms. However, there is also the additional problem that if the first R Group is aromatic and the second is aliphatic, the matching of the second R Group must take this into account and not discount the possibility of a match because two of the atoms under consideration at the point of fusion are aromatic. The lists of atoms that match the second Fused R Group are merged with the first lists, including all combinations of one from each list. The merged list is processed to discard those that do not contain a fused ring system, and the algorithm proceeds as for other groups using the surviving merged lists.

RESULTS AND DISCUSSION

We describe here three distinct uses to which the Periscope system has been put.

Free–Wilson-Style Analyses. In 1964, Free and Wilson[25] published a method of analyzing the contribution to biological activity of particular fragments at particular positions around the core of a homologous series of compounds. The method has become applied widely over the intervening decades.[26] It involves identifying a substructure (the "core") common to a group of compounds and identifying the substituents at fixed positions about the core. Each substituent at a position becomes a descriptor for the parent molecule, and a regression equation is developed using these descriptors. One problem with this is that it is frequently the case that a particular substituent occurs only a small number of times, and there are a large number of different substituents. The result is that there are more descriptors than dependent variables, and a model, if it can be built at all, is unreliable. A Markush representation of the molecules can reduce this problem by breaking the substituents down into more than one piece and also by generalizing fragments so that, for example, methylene, ethylene, and propylene linkers are brought together into the single class alkylene. Having developed a Markush description for the compound set, i3am can be used to generate automatically the descriptors for both the training and test sets and any further prospective sets.

We demonstrate the use of the Periscope system for Free–Wilson analysis with two sets of data. In neither case do we claim that the model we produce is a good one; indeed, it is more than likely that there is overfitting in both cases, something that might be controlled for by, for example, splitting the data into training and test sets or leave-one-out validation. The purpose of the examples is merely to show what is possible using the system.

The first exemplary data are a set of compounds active against the B-RAF kinase and recently used by Yang et al. for a 3D-QSAR study. This data set was selected because it was recent and the authors have provided the 2D structures and IC_{50} data for 59 compounds as part of the Supporting Information (their Table S1). Their table contains data for 61 compounds, but the last two are not printed correctly, so it is not possible to say what the structures are. The 59 structures were converted to SMILES strings, and two Markush descriptions generated. A schematic diagram of the Markush structure is shown in Figure 10, and the full MIL files are provided in the Supporting Information. The first encoding was a specific one in which each different R Group was described by an exact substructure. The output file from i3am in this case was the textual description of the R Group, a portion of which being shown in Table 1 and the full file being provided in the

dx.doi.org/10.1021/ci3000387 | J. Chem. Inf. Model. 2012, 52, 1936–1947

Table 1. Sample Output from Markush Analysis of Compounds for Free–Wilson-Style Analysis, Using Precise R Group Specification and Textual Descriptions[a]

Molecule	Amide_Linker	L	R1	RingC_4	RingC_5
1	Amide_Linker_C	L_Nitrogen	R1_Nitrogen	RingC_4_Cl	RingC_5_Hydrogen
2	Amide_Linker_C	L_Nitrogen	R1_Hydrogen	RingC_4_Hydrogen	RingC_5_Chlorine
9	Amide_Linker_C	L_Nitrogen	R1_Hydrogen	RingC_4_Hydrogen	RingC_5_Morpholine
11	Amide_Linker_C	L_Nitrogen	R1_Alkyl	RingC_4_Cl	RingC_5_Hydrogen
18	Amide_Linker_Thio	L_Nitrogen	R1_Hydrogen	RingC_4_Cl	RingC_5_Hydrogen
19	Amide_Linker_C	L_Bond	R1_Hydrogen	RingC_4_Cl	RingC_5_Hydrogen
24	Amide_Linker_S	L_Bond	R1_Hydrogen	–	–
55	Amide_Linker_C	L_Nitrogen	R1_Hydrogen	RingC_4_Hydrogen	RingC_5_Hydrogen

[a]The table is truncated both in number of lines and columns. The full table is given in the Supporting Information.

Table 2. Sample Output from Markush Analysis of Compounds for Free–Wilson-Style Analysis, Using General R Group Specification and SMILES Descriptions[a]

Molecule	Amide_Linker	L	R1	RingC	RingC_5	RingC_6
1	C(=O)N[Xe]	N[Xe]	–	–c1ccc(cc1)[Xe]	–	–
2	C(=O)N[Xe]	N[Xe]	–	c1ccc(cc1)[Xe]	Cl[Xe]	–
9	C(=O)N[Xe]	N[Xe]	–	–c1ccc(cc1)[Xe]	C1COCCN1[Xe]	–
11	C(=O)N[Xe]	N[Xe]	C[Xe]	c1ccc(cc1)[Xe]	–	–
18	C(=S)N[Xe]	N[Xe]	–	c1ccc(cc1)[Xe]	–	–
19	C(=O)N[Xe]	–	–	c1ccc(cc1)[Xe]	–	–
24	N(S(=O)=O)[Xe]	–	–	c1 cm^3(sc1)[Xe]	–	–
55	C(=O)N[Xe]	N[Xe]	–	c1ccc(cc1)[Xe]	–	F[Xe]

[a]The table is truncated both in number of lines and columns. The full table is given in the Supporting Information. The xenon atom indicates where the substituent was attached to the parent moiety.

Supporting Information. The second Markush description was more general at several positions, and the output file from i3am, shown in Table 2, is the SMILES strings of the relevant portions of the molecules, with a xenon atom marking the point of attachment to the parent moiety. The two encoding methods contain the same information, but the latter is quicker to develop and demonstrates some of the flexibility in the system. This example does not show the use of the Markush to provide general descriptors for parts of the molecule, such as alkylene linkers as described above, as the molecules in the series did not lend themselves to it.

A Free–Wilson model was developed using the Multiple Linear Regression module of SAS's JMP program,[28] using the text strings in each column of data as categorical descriptors. The plot of actual versus predicted IC_{50} for the full set of 59 molecules is shown in Figure 11. The Root Mean Square Error (RMSE) is 0.25, and the R^2 is 0.95, which are comparable to the ones Yang et al. (R^2 of between 0.92 and 0.96) produced for their 3D QSAR models. We make no claims here for the superiority or utility of one model or technique over the other but merely wish to demonstrate that the Markush breakdown of the molecules can be used to generate Free–Wilson-style models of acceptable quality.

As a second example, we have encoded the Markush structure from patent application WO2009/016462(A2).[29] This is a patent application published by Pfizer for compounds that inhibit the enzyme Diacylglycerol O-acyltransferase 1 (DGAT-1), an enzyme of interest in the treatment of diabetes. We have encoded the Markush described in the section of the application entitled "Summary of the Invention" starting on page 3. The structures of 54 example compounds given were extracted from the GVKBio database[30] along with their quoted activities and queried with the MIL. A Free–Wilson model was generated from the descriptors so generated using the textual

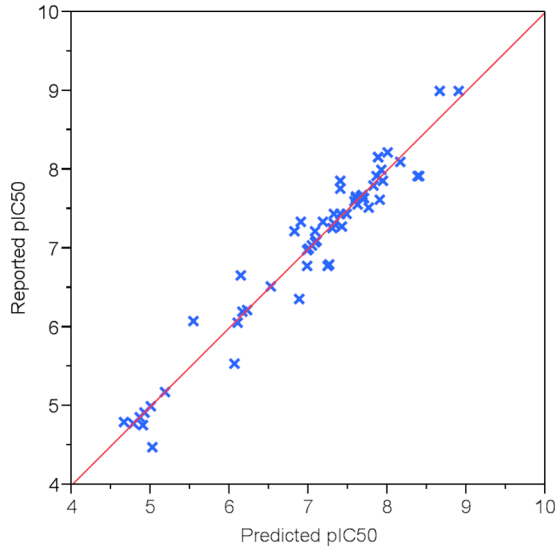


Figure 11. Plot of actual vs predicted activity from the Free–Wilson-style QSAR model derived from the Markush analysis of Yang et al.

output format rather than the SMILES. Figure 12 shows a plot of the predicted activities against those calculated by the Free–Wilson model. The RMSE is 0.28, and the R^2 0.71. By contrast, Figure 13 shows a similar plot for a model built with more conventional QSAR descriptors (calculated logP, ACDlogP/D, molecular weight, hydrogen bond acceptor and donor counts, molecular volume, polar- and nonpolar-surface areas, number of rotatable bonds, ring count, predicted ionization state at pH 7.4, and Lipinski count). The model is clearly less good and has an RMSE of 0.43 and an R^2 of 0.28. The MIL file for the Markush and the datafiles for the two models are included in the Supporting Information.

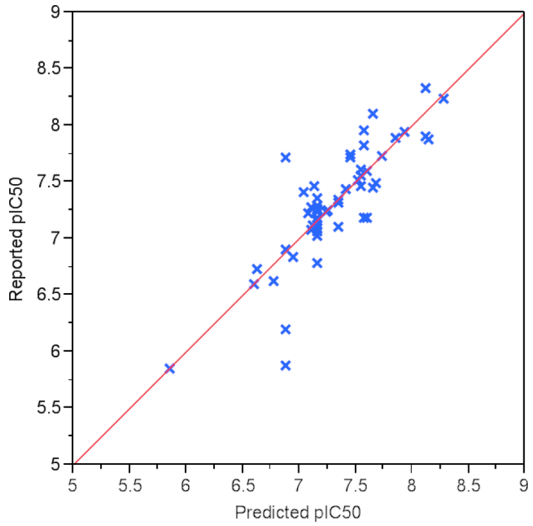


Figure 12. Plot of measured versus calculated pIC_{50} values for DGAT-1 inhibitors using a Free–Wilson model generated from the Markush in patent WO2009/016462(A2). The RMSE is 0.28, and the R^2 is 0.71.

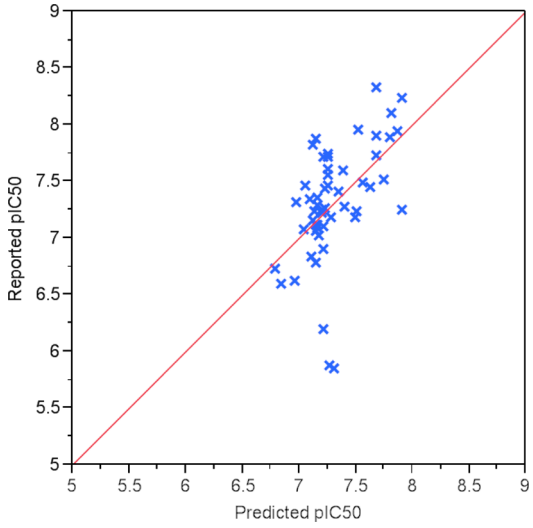


Figure 13. Plot of measured versus calculated pIC_{50} values for DGAT-1 inhibitors using a variety of "conventional" QSAR descriptors as described in the text. The RMSE is 0.43, and the R^2 is 0.28.

Monitoring Controlled Substances in Compound Collections. The UK Home Office, in common with most governments of the world, proscribes the manufacture, trade in, and shipping of a number of chemical structures, most notably chemical weapons and psychoactive substances ("illegal drugs"). Where they can be kept at all, such compounds must usually be held in secure storage and individual uses recorded. Where they must be transported, a license is usually required on each occasion. As well as proscribing individual compounds, the UK legislation currently includes some 19 Markush structures. In addition, a number of the individual compound definitions include "any ethers and esters." An example of such a Markush structure for phenethylamine is shown in Figure 14. At AstraZeneca, we had an alerting system that used a number of substructure searches run with Accelrys' ISIS chemoinformatics package, with the searches being run on a monthly basis. The somewhat general nature of the Markush

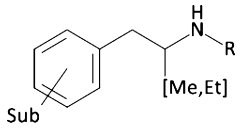


Figure 14. Markush structure for phenethylamine controlled substances. [Me,Et] is methyl or ethyl, R is alkyl, and Sub is alkyl, alkoxy, alkylenedioxy, or halogen. There must be at least one Sub; [Me,Et] and R are optional. Alkyl is C_1-C_6, branched or unbranched, acyclic but allowing cyclopropyl. Alkoxy is an oxygen atom with an attached alkyl, and alkylenedioxy is two alkoxy groups that come together to form a ring.

structures meant that the substructures often produced a large number of hits, which then had to be checked manually. The phenethylamine substructures produced some tens of thousands of hits in the AstraZeneca compound database, more than could feasibly be checked in this manner, with the result that a large number of compounds had their uses unnecessarily restricted. When the Markush was encoded as a MIL file and searched with i3am, there were in fact only 53 hits. The rest were thus released for general use, giving a significant increase in the number of compounds available for screening. All the proscribed Markush structures have now been encoded, and every night all new registrations to the compound database are checked against them. The new system is more precise and requires much less human intervention than the previous one, saving both time and money. It retains the accuracy of the previous system, in terms of the correctly identified controlled substances, so-called true positives, but reduces dramatically the number of false positives. We have also made an online tool available internally so that, for example, library designs can be screened for controlled substances before being submitted for synthesis, either internally or at third-party organisations, thus saving on the cost of making compounds that will never be used.

The MIL file for the phenethylamine search is given in the Supporting Information. To show the benefit of the Markush search over a single substructure search, the snapshot of PubChem from November 2011 was searched by both methods. The SMARTS string c1ccccc1[CH2;!R][$([CH2;!R]),$([CH;!R][CH3]),$([CH;!R][CH2][CH3])][$([NH](C)C),$([NH2])] was used for the substructural query. In a single SMARTS pattern, it is not possible to specify that there must be at least one carbon, oxygen, or halogen atom on the phenyl ring, so this SMARTS will certainly give false positives in which the phenyl is not substituted or is substituted with inappropriate groups such as amines. The substructure search gave 221192 hits. Four examples of false positive hits are shown in Figure 15. From the description above, they are clearly not correct matches to the controlled substance: three have incorrect groups on the nitrogen atom, and the other has a fused ring on the phenyl containing incorrect atoms.The Markush search gave only 1213 hits, 4 examples being shown in Figure 16. These are much simpler molecules, all of which fit the appropriate description. On a reasonably modern single processor Linux workstation, the substructure search took 50 min and the Markush search 11 h 7 min. Clearly, the Markush search is much more time consuming but still manageable overnight and, therefore, in our opinion, worth the wait for the dramatic reduction in false positive hits.

Searching a Very Large Virtual Library. Hu et al. have described[31] a system for generating and searching very large virtual libraries on the basis of chemistries developed for

dx.doi.org/10.1021/ci3000387 | J. Chem. Inf. Model. 2012, 52, 1936–1947

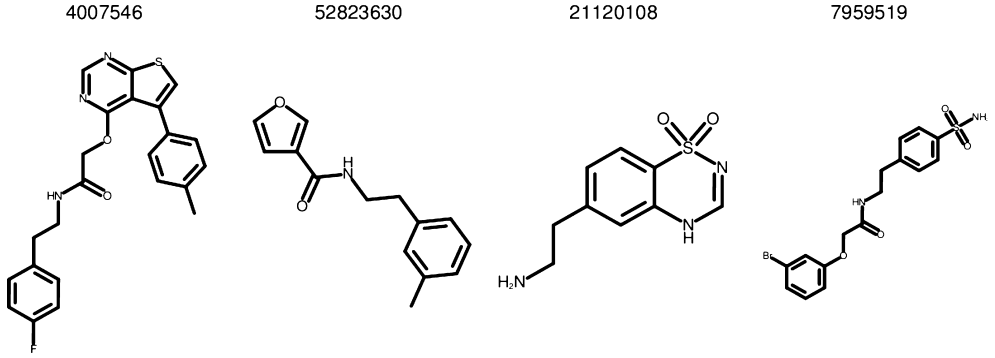


Figure 15. Examples of false positive hits for the SMARTS-based phenethylamine substructure search of PubChem.

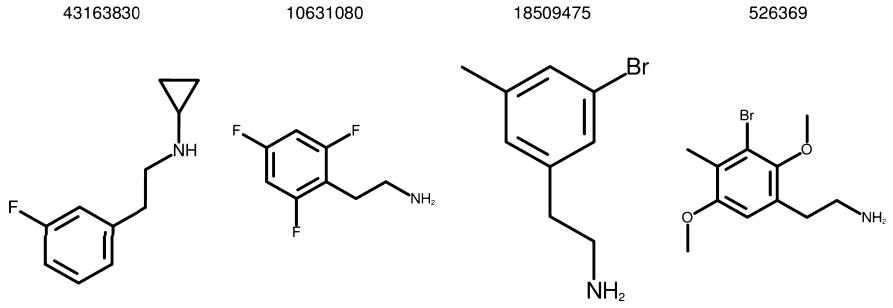


Figure 16. Examples of correct hits for the Markush search of PubChem.

smaller-scale libraries. In such an approach, the libraries generally consist of a fixed core with between 1 and 4 reaction positions, with a small number of reagents at each position, leading to each library comprising of a few hundred to a thousand molecules. More often than not, the number of reagents that could have been used at each position is in the hundreds or thousands, such that the number of compounds that could have been synthesized using the same reaction and all available reagents for a particular library might be in the order of many millions. Multiplying this by the number of libraries, one potentially has a virtual library of the order of 10^{12} molecules, although not all of them are necessarily unique. Clearly, it is not feasible to enumerate and search that number of compounds, but the problem becomes tractable if the libraries are maintained in Markush format. Then, to see if a query compound is in a particular library, one matches the core against the molecule, and if it is found, the actual substituents in the query are compared with the possible substituents derived from the reagents used to create the library. In this example, the full power of Periscope's general Markush searching is not required, and indeed both Digital Chemistry's Torus and ChemAxon's Marvin systems are already capable of being used in this way. Nonetheless, it was relatively straightforward to encode each possible library by a MIL file such that exact and substructure searches may be done using i3am. Without any attempts at optimization, a query set of a few hundred molecules can be searched against a full set of some 4000 libraries in Markush format in about 8 h on a single workstation. The problem is embarrassingly parallel,[32] so that if spread across a cluster of machines, it scales almost linearly making very fast searching possible.

As well as finding exact matches between a query structure and a library molecule, i3am can identify partial matches, either where a library molecule is a subset of the query molecule or the query molecule is a subset of a library molecule. In the former case, this is where a library molecule has an additional group not found in the query, but every atom in the query matches an atom in the library molecule. In the latter case, a part of the query molecule fails to match any of the R Groups at one or more positions in the Markush. If either of these occurs, a separate step is performed in which the library is partially enumerated to generate a set of molecules containing all combinations of possibilities for the unmatched R Groups. The Rascal[33] maximum common edge substructure (MCES) algorithm is then used to find the closest match to the query molecule. Having found a virtual library that contains the query molecule or something very similar to it, it is straightforward to synthesize a new library in order to expand the chemical space around the molecule of interest.

■ CONCLUSION

We have described a system for encoding and searching chemical Markush structures, and three uses thereof, namely the generation of Free–Wilson analyses for chemical series, the monitoring of controlled chemicals and the querying of large virtual libraries in order to find smaller libraries for synthesis to expand chemical equity around a compound of interest. All three uses have been of assistance in drug discovery projects at AstraZeneca. We are also currently investigating its value as a highly flexible substructure searching tool.

Deng et al.[10] commented recently that "For Markush structure representation, a more commonly adopted format is preferred." We believe the MIL format described herein should be that representation. One motivation for publishing the format in full detail is to encourage its adoption. It would then be possible to build, for example, an open searchable archive of Markush structures from patents for use by researchers working in neglected diseases areas.

■ ASSOCIATED CONTENT

Supporting Information

The DOI of the source paper for Yang et al. is ci100427j. The following items are available: (1) **US1506316.milf**: A MIL file for Markush' original patent. (2) **WO2009_016462.milf**: A MIL file for DGAT-1 patent WO2009/016462. (3) **WO2009016462_free_wilson_data.txt**: Data file for Free–Wilson model of structures in DGAT-1 patent WO2009/016462. (4) **WO2009016462_conventional_qsar_data.txt**: Data file for QSAR model of structures in DGAT-1 patent WO2009/016462. (5) **SI_1_MIL_Description.pdf**: A full description of the Markush Input Language. (6) **phenethylamine.milf**: The MIL file for the phenethylamine controlled substance Markush search. (7) **ci100427j.smi**: The SMILES strings for the structures used in the Free–Wilson-style analysis. (8) **ci100427j.data**: The corresponding activity values. (9) **ci100427.milf**: The MIL file for the Markush used in the Free–Wilson-style analysis, with specific R Group descriptors, used to create Table 1. (10) **ci100427_general.milf**: The MIL file for the Markush used in the Free–Wilson-style analysis, with general R Group descriptors, used to create Table 2. (11) **table1_full.txt**: The full data for Table 1. (12) **table2_full.txt**: The full data for Table 2. (13) **mil_schema.xsd**: A schema for the MIL, generated automatically from a selected sample of MIL files using the program trang.[34] This material is available free of charge via the Internet at http://pubs.acs.org/.

■ AUTHOR INFORMATION

Corresponding Author

*E-mail: david.cosgrove@astrazeneca.com.

Notes

The authors declare no competing financial interest.

■ ACKNOWLEDGMENTS

We are grateful to a large number of chemists for helpful discussions but particularly Ed Griffen. Daniel Taylor, Stuart Bowden, and Richard Kilburn are thanked for their advice on controlled substance specification and searching. Thanks are also due to Suzanne Pears for her persistent testing of early versions of Menguin. Finally, we thank Sorel Muresan, Jin Li, and the anonymous reviewers for their careful reading and helpful comments on an earlier draft of the manuscript.

■ REFERENCES

(1) Markush, E. A. Pyrazolone Dye and Process of Making the Same. U.S. Patent US1506316, 1924.

(2) Lynch, M. F.; Holliday, J. D. The Sheffield Generic Structures Project: A retrospective review. *J. Chem. Inf. Comput. Sci.* **1996**, *36*, 930–936.

(3) Ebe, T.; Sanderson, K. A.; Wilson, P. S. The Chemical Abstracts Service generic chemical (Markush) structure storage and retrieval capability. *J. Chem. Inf. Comput. Sci.* **1991**, *31*, 31–36.

(4) Chemical Abstracts Service. http://www.cas.org (accessed April 17, 2012).

(5) Benichou, P.; Klimczak, C.; Borne, P. Handling genericity in chemical structures using the Markush DARC software. *J. Chem. Inf. Comput. Sci.* **1997**, *37*, 43–53.

(6) Thomson Reuters. http://www.thomsonreuters.com (accessed April 17, 2012).

(7) ChemAxon Kft. http://www.chemaxon.com (accessed April 17, 2012).

(8) Digital Chemistry. http://www.digitalchemistry.co.uk (accessed April 17, 2012).

(9) Barnard, J. M.; Wright, P. M. Towards in-house searching of Markush structures from patents. *World Patent Inf.* **2009**, *31*, 97–103.

(10) Deng, W.; Berthel, S. J.; So, W. V. Intuitive patent Markush structure visualisation tool for medicinal chemists. *J. Chem. Inf. Model.* **2011**, *51*, 511–520.

(11) Downs, G. M.; Barnard, J. M. Chemical patent information systems. *Wiley Interdiscip. Rev.: Comput. Mol. Sci* **2011**, *1*, 727–741.

(12) Marvin. http://www.chemaxon.com/products/markush-ip (accessed April 17, 2012).

(13) ChemProspector. http://infochem.de/news/projectdisplay.shtml?chemprospector.shtml (accessed May 15, 2012).

(14) Murray-Rust, P.; Rzepa, H. S. Chemical markup, XML, and the Worldwide Web. 1. Basic principles. *J. Chem. Inf. Comput. Sci.* **1999**, *39*, 928–942.

(15) Adams, N.; Winter, J.; Murray-Rust, P.; Rzepa, H. S. Chemical markup, XML and the Worldwide Web. 8. Polymer markup language. *J. Chem. Inf. Model.* **2118**, *48*, 2118–2128.

(16) SMILES Arbitrary Target Specification. http://en.wikipedia.org/wiki/Smiles_arbitrary_target_specification (accessed April 5, 2012).

(17) Accelrys, Inc. http://accelrys.com (accessed April 5, 2012).

(18) MDL CTfile Formats. http://accelrys.com/products/informatics/cheminformatics/ctfile-formats/no-f (accessed July 12, 2012).

(19) PubChem. http://pubchem.ncbi.nlm.nih.gov (accessed November 2011).

(20) *JDraw*, v1.1.200.121; Accelrys, Inc.: San Diego, CA, 2010.

(21) *Flex*, v4.6; Adobe Systems Incorporated: San Jose, CA, 2012.

(22) *Lexichem*, v2.1.0; OpenEye Scientific Software: Santa Fe, NM, 2011.

(23) *OEChem*, v1.7.4; OpenEye Scientific Software: Santa Fe, NM, 2012.

(24) Carpaneto, G.; Martello, S.; Toth, P. Algorithms and codes for the assignment problem. *Ann. Oper. Res.* **1988**, *13*, 193–223.

(25) Free, S. M.; Wilson, J. W. A mathematical contribution to structure–activity studies. *J. Med. Chem.* **1964**, *7*, 395–399.

(26) Patel, Y.; Gillet, V. J.; Howe, T.; Pastor, J.; Oyarzabal, J.; Willett, P. Assessment of additive/nonadditive effects in structure–activity relationships: Implications for iterative drug design. *J. Med. Chem.* **2008**, *51*, 7552–7562.

(27) Yang, Y.; Qin, J.; Liu, H.; Yao, X. Molecule dynamics simulation, free energy calulation and structure–based 3D-QSAR studies of B-RAF kinase inhibitors. *J. Chem. Inf. Model.* **2011**, *51*, 680–692.

(28) *JMP* v8; SAS Institute, Inc.: Cary, NC, 2008.

(29) Dow, R. L.; Munchfor, M. J. Substituted Bicyclolactam Compounds. Patent WO2009/016462 A2, Pfizer Global Research and Development, 2009.

(30) Jagarlapudi, S. A. R. P.; Kishan, K. V. R. *Chemogenomics: Methods and Applications*; Humana Press: New York, 2009; Vol. *575*, pp 159–172.

(31) Hu, Q.; Peng, Z.; Kostrowicki, J.; Kuki, A. Methods in Molecular Biology. In *Chemical Library Design*; Zhou, J. Z., Ed.; Humana Press: New York, 2011; Vol. *685*, pp 253–276.

(32) Embarassingly Parallel. http://en.wikipedia.org/wiki/Embarrassingly_parallel (accessed April 17, 2012).

(33) Raymond, J. W.; Gardiner, E. J.; Willett, P. RASCAL: Calculation of graph similarity using maximum common edge subgraphs. *Comput. J.* **2002**, *45*, 631–644.

(34) *Trang*, v20091111; Thai Open Source Software Center, Ltd.. http://code.google.com/p/jing-trang (accessed July 12, 2012).