HEURISTICS FOR SEARCHING CHEMICAL STRUCTURES

A Thesis presented to the Faculty of the Graduate School
University of Missouri-Columbia

In Partial Fulfillment
of the Requirements for the Degree
Master of Science

by
NANDINI BASU

Dr. Toni Kazic, Thesis Supervisor

MAY 2007
The undersigned, appointed by the Dean of the Graduate School, have examined the thesis entitled

HEURISTICS FOR SEARCHING CHEMICAL STRUCTURES

presented by Nandini Basu

A candidate for the degree of Master of Science

And hereby certify that in their opinion it is worthy of acceptance.

Dr. Toni Kazic

Dr. Chi-Ren Shyu

Dr. Mary Schaeffer
ACKNOWLEDGMENTS

I am deeply indebted to my research advisor Dr. Toni Kazic for giving me an opportunity to resume my research in and complete my Computer Science MS program. Without her guidance, support and encouragement this would not have been possible.

I would like to thank Dr. Mary Polaco and Dr. Shyu for serving in my committee and helping me in my thesis.

I would like to thank the Klotho team for helping me with structures. And also the other team members Amith, Archana, Avanthi, Bill, Deepthi, Gaurav, Raman, Jiahui, Fatten, Meeta and Michael.

My friends made a difficult journey lot of fun and much easier, especially Cathy, Charu, Deepak, Karan, Karthik, Paco, Nikunj, Raj, Shiva, Shraddha, Sundeep and Vibha.

All my relatives gave me constant encouragement and I would like to take this opportunity to thank them.

And above all, I would like to thank my father Prabhat Kumar Basu, my mother Minoti Basu, my brother Pradipta Kumar Basu and my sister-in-law Sarmistha Basu. Without their patience, support and encouragement and love I would not have achieved this goal in my life.
# TABLE OF CONTENTS

ACKNOWLEDGMENTS ................................................................. ii

LIST OF FIGURES ...................................................................... v

LIST OF TABLES ................................................................. vii

1 Introduction ................................................................. 1
   1.1 Motivation ............................................................ 1
   1.2 Chemistry behind search methods ................................. 1
      1.2.1 Isomers ..................................................... 2
      1.2.2 Constitutional or Structural isomers ....................... 3
      1.2.3 Configurational Isomers ................................... 6
   1.3 Existing Structure Search Methods ............................... 10

2 Materials ................................................................. 13
   2.1 Platforms and Languages .......................................... 13
   2.2 Klotho ............................................................ 13
   2.3 Config Rule ...................................................... 14
   2.4 Term Forms ...................................................... 16

3 Search Heuristics and Algorithms ...................................... 18
   3.1 Descriptors ....................................................... 18
   3.2 Isomer Search Algorithms ....................................... 19
   3.3 Substructure Search Algorithm ................................ 22
   3.4 Isoconnected Structure Search Algorithm ................. 23

4 Validation and Benchmarking of Search Procedures ............ 27
   4.1 Isomer Search ................................................... 27
   4.2 Substructure Search ............................................. 37
   4.3 Isoconnected Structure Search ................................ 41
   4.4 General Queries ................................................ 41
LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Example of aliphatic, cyclic and aromatic compounds</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>Example of substructure: Pyrimidine is a substructure of purine</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>Classification of Isomers</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>Skeletal isomers</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>Positional isomers</td>
<td>4</td>
</tr>
<tr>
<td>6</td>
<td>Functional isomers</td>
<td>5</td>
</tr>
<tr>
<td>7</td>
<td>Keto-enol tautomers</td>
<td>5</td>
</tr>
<tr>
<td>8</td>
<td>Ring-chain Tautomers</td>
<td>5</td>
</tr>
<tr>
<td>9</td>
<td>Chiral and Achiral centers</td>
<td>6</td>
</tr>
<tr>
<td>10</td>
<td>Geometric isomers</td>
<td>7</td>
</tr>
<tr>
<td>11</td>
<td>Enantiomers</td>
<td>7</td>
</tr>
<tr>
<td>12</td>
<td>Diastereomers</td>
<td>8</td>
</tr>
<tr>
<td>13</td>
<td>Epimers</td>
<td>8</td>
</tr>
<tr>
<td>14</td>
<td>Meso-isomers</td>
<td>9</td>
</tr>
<tr>
<td>15</td>
<td>Anomers</td>
<td>9</td>
</tr>
<tr>
<td>16</td>
<td>Ethanol</td>
<td>14</td>
</tr>
<tr>
<td>17</td>
<td>Algorithm 1 for structural isomer search</td>
<td>20</td>
</tr>
<tr>
<td>18</td>
<td>Algorithm 2 for configurational isomer search</td>
<td>21</td>
</tr>
<tr>
<td>19</td>
<td>Term form information for D-mannose and D-glucose for isomer differentia-</td>
<td>22</td>
</tr>
<tr>
<td></td>
<td>tion</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>Algorithm 3 for substructure search</td>
<td>23</td>
</tr>
<tr>
<td>21</td>
<td>Term form information for amine and ethylamide for substructure search</td>
<td>24</td>
</tr>
<tr>
<td>22</td>
<td>Algorithm 4 for isoconnected structure search</td>
<td>25</td>
</tr>
<tr>
<td>23</td>
<td>Algorithm 5 for atom by atom matching</td>
<td>26</td>
</tr>
<tr>
<td>24</td>
<td>Flow diagram showing the use of pre-computed descriptor information in</td>
<td>29</td>
</tr>
<tr>
<td></td>
<td>various structure search methods</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>Flow chart for functional isomer search</td>
<td>29</td>
</tr>
<tr>
<td>26</td>
<td>Flow chart for positional isomer search</td>
<td>30</td>
</tr>
<tr>
<td>27</td>
<td>Flow chart for skeletal isomer search</td>
<td>31</td>
</tr>
<tr>
<td>Figure</td>
<td>Page</td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>------</td>
<td></td>
</tr>
<tr>
<td>28. Flow chart for tautomer search</td>
<td>32</td>
<td></td>
</tr>
<tr>
<td>29. Flow chart for geometrical isomer search</td>
<td>33</td>
<td></td>
</tr>
<tr>
<td>30. Flow chart for diastereomer search</td>
<td>34</td>
<td></td>
</tr>
<tr>
<td>31. Flow chart for enantiomer search</td>
<td>35</td>
<td></td>
</tr>
<tr>
<td>32. Flow chart for meso-isomers search</td>
<td>36</td>
<td></td>
</tr>
<tr>
<td>33. Flow chart for substructure search</td>
<td>39</td>
<td></td>
</tr>
<tr>
<td>34. Plot for comparison of different descriptors. Data taken from Table 8</td>
<td>40</td>
<td></td>
</tr>
<tr>
<td>35. Plot for comparison of different combinations of descriptors. Data taken from Table 9</td>
<td>41</td>
<td></td>
</tr>
<tr>
<td>36. Flow chart for the isoconnectivity search method</td>
<td>42</td>
<td></td>
</tr>
<tr>
<td>37. Verification of isomer search method</td>
<td>43</td>
<td></td>
</tr>
<tr>
<td>38. Verification of substructure search method</td>
<td>46</td>
<td></td>
</tr>
<tr>
<td>39. Screenshot of user interface of main structure search page</td>
<td>47</td>
<td></td>
</tr>
<tr>
<td>40. Screenshot of user interface of substructure search page</td>
<td>47</td>
<td></td>
</tr>
<tr>
<td>41. Screenshot of user interface of functional isomer search form</td>
<td>47</td>
<td></td>
</tr>
<tr>
<td>42. Screenshot of result for structural isomers search</td>
<td>48</td>
<td></td>
</tr>
<tr>
<td>43. Screenshot of result for substructure search</td>
<td>48</td>
<td></td>
</tr>
<tr>
<td>44. Overview of all the structure search methods combined together</td>
<td>51</td>
<td></td>
</tr>
<tr>
<td>45. Algorithm 6 for functional isomer search</td>
<td>55</td>
<td></td>
</tr>
<tr>
<td>46. Algorithm 7 for tautomer search</td>
<td>56</td>
<td></td>
</tr>
<tr>
<td>47. Algorithm 8 for skeletal isomer search</td>
<td>57</td>
<td></td>
</tr>
<tr>
<td>48. Algorithm 9 for positional isomer search</td>
<td>58</td>
<td></td>
</tr>
<tr>
<td>49. Algorithm 10 for geometric isomer search</td>
<td>59</td>
<td></td>
</tr>
<tr>
<td>50. Algorithm 11 for meso-isomer search</td>
<td>60</td>
<td></td>
</tr>
<tr>
<td>51. Algorithm 12 for enantiomer search</td>
<td>61</td>
<td></td>
</tr>
<tr>
<td>52. Algorithm 13 for diastereomer search</td>
<td>62</td>
<td></td>
</tr>
</tbody>
</table>
# LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Isomer descriptions</td>
<td>10</td>
</tr>
<tr>
<td>2. Name and description of primary and secondary descriptors</td>
<td>19</td>
</tr>
<tr>
<td>3. Types of isomer searches included in the isomer search method</td>
<td>22</td>
</tr>
<tr>
<td>4. Compound type distribution in Klotho database</td>
<td>27</td>
</tr>
<tr>
<td>5. Main query table. These queries are implemented in the front end</td>
<td>28</td>
</tr>
<tr>
<td>6. Isomer search data</td>
<td>37</td>
</tr>
<tr>
<td>7. Additional structure search queries</td>
<td>38</td>
</tr>
<tr>
<td>8. Comparison of CPU runtime for substructure search of purine ((\text{C5H4N4})) without and with one descriptor</td>
<td>39</td>
</tr>
<tr>
<td>9. Comparison of CPU runtime for substructure search of purine ((\text{C5H4N4})) with different combinations of descriptors</td>
<td>40</td>
</tr>
<tr>
<td>10. Verification of some isomer search queries</td>
<td>44</td>
</tr>
<tr>
<td>11. Verification of some substructure search and exact structure search queries</td>
<td>45</td>
</tr>
<tr>
<td>11. Symbol table for algorithms</td>
<td>55</td>
</tr>
</tbody>
</table>
1 Introduction

1.1 Motivation

Chemical and biochemical compounds form the building blocks of most things essential in life. Databases containing chemical and biochemical compounds are being developed continuously to keep at par with the developing research and industry in the chemical, biochemical and pharmaceutical fields. Apart from efficient storing of these compounds as structures, it is also highly essential for such databases of any size to have a search tool for easy access and retrieval of the information. Efforts are made to create optimum search tools which will ensure a complete and quick search of the entire database.

This thesis explores the development of an intelligent structure search tool that implements different types of structure search methods in our database, Klotho [14]. There are four different structure search methods implemented in this search tool. They are isomer search, substructure search, isoconnected structure search and general structure search. Isomer search method retrieves the structural isomers and the configurational isomers of the query structure. Substructure search method retrieves those compounds which contain the query structure as a part of their whole structure. Isoconnected structure search method retrieves those compounds whose structure is isoconnected with that of the query structure. This method also enables exact structure search, retrieving compounds with same stereochemistry. The general structure search method retrieves certain chemical propert information of chemical compounds.

1.2 Chemistry behind search methods

A brief overview of the chemistry involved in the search methods is necessary at this point for a better understanding of the research investigation. Chemical compounds consist of various atoms joined to each other by different types of bonds. Different combinations of such bonds give rise to different types of structures which can be classified under some categories. A compound can be aliphatic or a chain structure. Compounds which contains cycles which are rings with single bonds are classified as cyclic. Aromatic compounds have electrons moving freely around in a ring resulting in adjacent single and double bonds in the ring called a conjugated system. Figure 1 shows an
example of aliphatic, cyclic and aromatic compounds.

A substructure of a structure is any part of the whole structure. For e.g. in Figure 2, the structure of compound pyrimidine is part of the structure of compound purine. Hence pyrimidine is a substructure of purine.

1.2.1 Isomers

Every chemical compound or species has a distinct formula which distinguishes it from others. A molecular formula represent the total number of each distinct elements present in a compound or a molecule. Isomers are compounds that have the same molecular formula but are chemically or physically different. There are two broad classifications of isomers - constitutional isomers and stereo-isomers. Constitutional isomers, also known as structural isomers, differ in the connectivity of the constituent atoms. On the other hand, stereo-isomers have same atom to atom connectivity but differ in the arrangement of atoms in space. Stereoisomers can further be classified as configurational isomers and conformational isomers. Configurational isomers have the same groups bound to them but are different overall due to their orientation of the groups in space. Conformational isomers also known as non-configurational isomers have the same molecular formula and group connectivity but a different spatial arrangement due to the rotation of a single bond. Figure 3 shows a broad classification of all isomers. As evident from the figure there structural isomers and configurational
Figure 3: Classification of Isomers

can be further classified into distinct categories.

1.2.2 Constitutional or Structural isomers

Structural isomers have same molecular formula i.e. same number of each distinct atom. They are distinguished by different atom to atom connectivity and can be further classified according to the different types of connections. There are four main types of structural isomers: Skeletal isomers, Positional isomers, Functional isomers and Tautomers.

Skeletal isomers have the same functional group but different skeleton or backbone. In the following example as seen in Figure 4, ethylbenzene and 1,2-dimethylbenzene have different skeletons. In ethylbenzene an ethyl group (-CH$_2$-CH$_3$) is attached to carbon position 1 of benzene molecule whereas in 1,3-dimethylbenzene two methyl (-CH$_3$) groups are attached to position 1 and 3 of benzene molecule.

Positional isomers have the same backbone and the same functional group. But the functional
group is present at different positions of the carbon chain, hence the name. An example of positional isomers is shown in Figure 5 where both compounds have the same atom groups attached at different positions. In 1,2-dimethylbenzene the methyl groups are attached to positions 1 and 2 of benzene molecule while in 1,3-dimethylbenzene two methyl (-CH$_3$) groups are attached to position 1 and 3 of benzene molecule.

Functional isomers are compounds with the same molecular formula but different functional groups. Figure 6 shows an example of functional isomers. Dimethyl ether is an ether with an ether functional group containing a -C-O-C connectivity. Ethanol is an alcohol containing an alcohol functional group with a C-OH connectivity.

Tautomers are structural isomers that are interconvertible by a chemical reaction called tautomerization. This reaction results in the migration of a hydrogen atom accompanied by simultaneous migration of adjacent conjugated double bonds. The migration of H atom and double bond occurs as an exchange. The H atom shift can be two types mainly - a 1, 3-H shift or a 1, 5 - H shift. Tautomers are easily interconvertible and exist in equilibrium with each other. There are two main
types of tautomerization. In the first one tautomerization takes place without any change in the architecture of the compound. *Keto-enol tautomerization* as shown in Figure 7 is an example of this kind where tautomerization takes place either between two rings or two chains. The second type is *ring chain tautomerization* in which migrations result in conversion of a ring to chain or vice versa as shown in Figure 8.
1.2.3 Configurational Isomers

Configurational isomers have same atom to atom connectivity but they differ in the arrangement of atoms in space. As shown in Figure 3 there are different types of configurational isomers. A brief overview of the different types of configurational isomers follows shortly. When carbon atoms in stereoisomers have four different groups attached to it, then those atoms are called chiral centers. In Figure 9, all carbon atoms except carbon 1 and carbon 6 are chiral carbons as the four bonds of carbon are attached to four different chemical groups. If more than one bond is attached to same group the atom is called achiral.

Geometric isomers are molecules with same molecular formula but different orientation of constituent groups about a double bond. There are two types of geometric isomers. Cis isomers have identical groups on the same side of the molecule while trans isomers have identical groups on the opposite site of the molecule. Figure 10 shows a common example of the geometric isomers which are cis- and trans-but-2-ene. In the cis- isomer, the methyl groups attached to carbon 2 and carbon 3 are on the same side of the double bond. But in the trans- isomer the methyl groups attached to carbon 2 and carbon 3 are on opposite sides of the double bond.

Enantiomers are configurational isomers that are non-identical and mirror-symmetric for all atoms. They are optically active for example levo/dextro-rotatory and are non-superimposable and inverted only by breaking bonds. For example D-sorbose and L-3-sorbose. Here D- stands for dextro meaning that the plane of polarized light is rotated to the right and L stands for levo meaning plane of polarized light is rotated to the left. As seen in Figure 11, both the compounds are mirror images.
Diastereomers are configurational isomers that are not mirror images to each other. They have more than one chiral center and have identical configuration for at least one chiral center. In Figure 12, D-glucose and D-talose are not mirror images of each other. They have identical configuration at two chiral centers and different configuration at two chiral centers which makes them a pair of diastereomers.

Epimers are a special case of diastereoisomers where there is a difference for one and only one asymmetric center. For e.g. in Figure 13, D-glucose and D-mannose are epimers. The configuration for these two compounds is different only at the chiral carbon 2.

Meso-isomers are super-imposable mirror images which have more than one chiral center. They have two planes of symmetry; the usual mirror plane of reflection and a second plane perpendicular to it in the middle of the molecule. The asymmetric centers are distributed around this second place.
Figure 12: Diastereomers

Figure 13: Epimers
so that they are mirror inverses of each other. Hence optical rotations from the upper and lower half of the molecule cancel each other out, making the entire molecule chiral. Figure 14 shows an example of such meso-isomers where upper half of molecule is mirror image of lower half of molecule.

Anomers are configurational isomers where the molecule is cyclized and the difference in configuration is about the anomeric carbon only. In case of aldoses the anomeric carbon is C1 and for ketoses it is C2. e.g. sugar hemiacetal. As seen in Figure 15, the anomeric C1 configuration is different for both molecules. To summarize, a brief definition of all the different isomer types are also given in Table 1.
### Table 1: Isomer descriptions

<table>
<thead>
<tr>
<th>Isomer Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Structural</td>
<td>Different atom connectivity</td>
</tr>
<tr>
<td>Positional</td>
<td>Same carbon skeleton but different positions of functional group</td>
</tr>
<tr>
<td>Functional</td>
<td>Different functional groups</td>
</tr>
<tr>
<td>Skeletal</td>
<td>Different carbon skeleton but same functional group</td>
</tr>
<tr>
<td>Tautomers</td>
<td>Compounds undergoing tautomerisation</td>
</tr>
<tr>
<td>Configurational</td>
<td>Same atom connectivity, but different arrangement in space around chiral atoms</td>
</tr>
<tr>
<td>Geometrical</td>
<td>Compounds with different orientation around multiple bonds. Functional group on same side of double bond are cis while on opposite sides are trans</td>
</tr>
<tr>
<td>Meso</td>
<td>Compounds with a horizontal plane of symmetry within the compound</td>
</tr>
<tr>
<td>Enantiomers</td>
<td>Compounds which are mirror images of each other around chiral atoms</td>
</tr>
<tr>
<td>Diastereomers</td>
<td>Compounds with at least one chiral center orientation different than each other</td>
</tr>
<tr>
<td>Epimers</td>
<td>Compounds with different orientation around exactly one chiral atom</td>
</tr>
<tr>
<td>Anomers</td>
<td>Compounds with different orientation around anomeric carbon</td>
</tr>
</tbody>
</table>

### 1.3 Existing Structure Search Methods

Various types of search methods for chemical structures in chemical and biochemical databases have been approached for almost five decades [12]. Of these, substructure search method is one of the most important structure search till date. The first substructure search matching algorithm was proposed by Ray and Kirsch in 1957 [19]. Each structure had a corresponding connection table which stored information about every atom’s connections. A one-to-one correspondence for every atom was sought between the query structure and each database structure to find a suitable match. Once an atom was successfully mapped, the neighboring atoms for the mapped atom in both structures were matched similarly. This was known as atom by atom mapping. A backtracking algorithm was used to map every atom of the query structure to the target structure. When faced with multiple paths to choose from, a backtracking algorithm picks one path and walks through it. When it can proceed no more, it goes back to the previous successful decision point and takes an alternate path. Such an algorithm performs an exhaustive search by going through all possible paths. In 1965, Sussenguth [20] proposed the first set-reduction substructure search algorithm. A node value matrix, a bond value matrix and a connection matrix were created for each structure. Partition of node sets were done based on node properties like number of bonds, bond types etc. Further subdivision was done in the following relaxation process which involved a connectivity property for the nodes. But Jochelson et al [17] found a problem with the algorithm. They argued that the bond matrix did not indicate whether an atom was involved in more than one bond. Also the
connection matrix did not indicate the nature of bond each pair of atoms was connected with. This lack of necessary information would give erroneous results especially in cases of compounds with conjugated and adjacent double bonds. In 1971, Ming and Tauber [18] proposed a correction for the Sussenguth algorithm ambiguity and introduced a method of including the bonds as nodes in an extended structure form. But this inherently changed the basic nature of the algorithm by changing the matrices and also requiring bigger storage space for the expanded graphs. In 1976 Ullmann’s algorithm [21] was published. It involves a backtracking tree search and Boolean matrices for both query and file structure. The backtracking step is followed by a relaxation process. It has been sometimes mentioned as the most efficient substructure search algorithm.

Screening systems have also been employed to reduce search space by removing compounds which are not possible targets [11]. They usually involve indexing the compounds with search keys based on some structural features and in the initial step those compounds that do not match the keys of the query compound are eliminated. CAS used screens, which were coded and stored in lists, to reduce search space in the first step [15]. This was followed by an atom by atom matching step for both exact structure and substructure searches. An algorithm proposing parallelism to increase the speed up the atom by atom mapping process by getting rid of the backtracking step was proposed in 1984 by Wipke et al. MACSS [16] used screens which were stored in inverted lists. It performed full structure searches via a hash code. It also has provisions for stereochemistry or isomer recognition. DARC [16] uses FRELs (Fragments reduced to an Environment that is Limited) in the screening process followed by ABAM (atom by atom matching). DARC can conduct searches for full structure, substructure and stereochemically defined structures. In S4 [16], molecules are encoded to give N connection tables for N atoms. Coding is sorted, stored and indexed and is compact enough to use only 15 bytes of space. This search algorithm is pretty fast and can do full structure, substructure, sterochemical and tautomeric isomer search. Among other methods, clustering methods have also been applied to speed up the substructure method [13]. Each of the clusters was characterized by a substructure present in all of the molecules. Hence some of the molecules could be present in multiple clusters. Though this process enabled fast and efficient search, storing of structures multiple times consumes unnecessary space.

A lot of commercially available structure search systems offer some isomer search tools along with other search methods like exact and substructure search. But most of these methods provide
option for stereoisomer and tautomer search mainly. Some of the very popular structure search systems are PubChem [7], SciFinder [8], MDLCROSSFIRE [5], DiscoveryGate [3], STN Express [10], ChemIDPlus [2] all of which offer exact and substructure search. PubChem [7] structure search system searches the PubChem Compound database which contains more than 10 million structures. The structures in the database are pre-clustered and cross-referenced by similar structure and exact structure groups. The search system also allows the user to search for exact and substructure search along with stereoisomers and tautomers. Options are also available to search for compounds with certain chemical properties. SciFinder [8] structure search system offers substructure and exact structure search. The exact structure search outputs stereoisomers and tautomers along with exact structures. MSDChem software [6] provides option for stereoisomer searches along with substructure searches. The structure search system available on Fischer Scientific website [4] and the software ChemBank [1] offer options for only substructure search. STNExpress offers options for stereochemical search [10]. Infochem’s SpresiWeb [9] offers isomer search along with substructure search. The isomer search method can search geometric isomers, stereo isomers and isotopic isomers.

Hence we felt the need of an isomer search tool which can perform a detailed search for all kinds of isomers. This work explores the development of an intelligent structure search tool that implements different search methods for the various types of structural and configurational isomers. We believe that the structure search system developed in this work offers more options than the conventional structure search tools. To our knowledge the isomer search method is more detailed than any other available isomer search methods. The user can search for different types of isomers of a particular structure in our in-house database Klotho [14] or within a list of structures provided by the user. This tool also implements other important structure search methods like substructure search and isoconnectivity search. Substructure search method retrieves those compounds which contain the query structure as a part of their whole structure. Isoconnectivity search method is an improvement over the popular exact structure search method which does not handle stereochemistry. The user retrieves a truly exact structure as the query structure with identical stereochemistry. The user can also retrieve identical structures that do not match stereochemistry with the query structure. Another interesting addition is the general structure search which includes two types of search. The user can search for compounds with specific properties or can search for all properties of a particular compound. This was developed to aide the user in all the other searches but can also be used to retrieve chemical property information for structures.
2 Materials

2.1 Platforms and Languages

Quintus Prolog (version 3.4) was used to develop this work on a Sun E450 running Solaris 8.0. The front-end is in html and is generated by perl modules. The front end also uses javascript and perl to process the data the input data.

2.2 Klotho

Klotho, our in-house database for chemical and biochemical compounds is a declarative prolog database. Klotho is one of three components of the Moirai project. According to Greek mythology, Klotho, one of the three Fates, performed the action of combing wool and spinning the thread of life while the other Fates waited for her. Symbolically computational tools have been developed to generate and recognize the structures of molecules and groups of molecules. These tools then later on interact with the other elements of Moirai that specify reaction information to provide other searching tools of great efficiency.

Chemical compounds can be represented as graphs where the atoms or molecule groups form the nodes of the graphs. The bonds connecting the atoms are the edges in the graphs and can be of single, double or triple. The goal for Klotho is to develop correct representations for compound structures which also include stereo-chemical information so that the mechanisms involved in biochemical reactions can be described with desired efficiency. Klotho has two types of definitions for each molecule. The first definition is called a config-rule which is also termed as a high-level language. The second definition is a low-level language called a term form. A description of a molecule given by the high-level language is identical to the molecule's description in the low-level language, and vice versa. The high-level language or config rule enables greater emphasis on the structure's overall topology and is used to generate and search both molecules and classes of molecules. That specification is transformed by Klotho's machinery (a graph grammar) into the low-level language or term form which is Klotho's primary output. Detailed insight into both these structure rules follows in the next two subsections.
2.3 Config Rule

A config-rule describes the general architecture of a molecule, whether it is a chain, a ring, or a ring system. It also gives information about the different substituent groups or atoms in the molecule, the different types of bonds between these groups and atoms and the nature of connection between them. Config Rule 1 is an example of a config rule for the compound ethanol shown in Figure 16.

config(ethanol,[
   chain([
       oxy(1,hyd),
       car(1,hydd&hyd),
       car(2,hydd&hyd&hyd)]).
   ]).

Config Rule 1: Ethanol

In Klotho, config rules are defined using the following Prolog predicate, where the syntax is 'config(CompoundName,CompoundSpecifications). with the period is absolutely required. CompoundName is the name of the compound which must be enclosed in single right quotes if it contains non-alphanumeric characters or begins with a number or capital letter. For e.g. pyrimidine and purine can be without quotes, but 'D-glucose' and '2,3-biphospho-D-glycerate' must be included in quotes. CompoundSpecifications is a list of terms, with the syntax as [term1,term2,...,termN] which more specifically describes the architecture of the molecule. The three main architecture types are chain, ring and ring system. Chains are for branched acyclic molecules or moieties. Rings are for molecules or moieties with exactly one cycle or an aromatic ring. And ring systems are for molecules or moieties with multiple cycles or aromatic rings. CompoundSpecifications mainly consists of the list of each these atoms or moieties in the molecule. There are many correct ways to write the CompoundSpecifications. For e.g. the Config Rule 2 shown below is quite different from Config Rule 1 for ethanol with quite different CompoundSpecifications and both of them are right. A more complicated config-rule for the big molecule ATP is shown in Config Rule 3.

config(ethanol,[
   chain([hydroxyl],
   )].

14
Config Rule 2: Ethanol

```prolog
config('ATP', [
    substituent('D-1-dehydroxy-5-oxy-ribofuranosyl'),
    substituent(adenyl),
    substituent(triphosphoryl),
    linkage(from('D-1-dehydroxy-5-oxy-ribofuranosyl',car(1)),
           to(adenyl,nit(9)),
           up,single),
    linkage(from(triphosphoryl,pho(1)),
           to('D-1-dehydroxy-5-oxy-ribofuranosyl',attach_to([oxy,car(5)])),
           nil,single)
]).
```

```prolog
config('D-1-dehydroxy-5-oxy-ribofuranosyl',[ 
    ring([oxy, 
          anomeric(1,hyd),
          car(2,hyd&&hydroxyl),
          car(3,hyd&&hydroxyl),
          car(4,oxymethyl&&hyd)])]).
```

```prolog
config(adenyl,[ 
    model(adenine,[
          diff(nit(9,hyd),nit(9))])]).
```

```prolog
config(adenine,[
    model(purine,[
          diff(car(6,hyd),car(6,amine(10)))]]).
```

```prolog
config(purine,[
    ring_system([ 
        ring([car(6,hyd)&,
              car(5)&,
              car(4)&,
              nit(3)&,
              car(2,hyd)&,
              nit(1)&]),
        ring([nit(7)&,
              car(8,hyd)&,
              nit(9,hyd)&,
              car(4)&,
              car(5)&]),
        conjugate(1,pseudopos([car(4),car(5)]),2,pseudopos([car(4),car(5)]))])].
```

Config Rule 3: ATP
2.4 Term Forms

Term-form or the low level language gives detailed information about every atom, atom number, isotope, charge, chirality. It also gives an adjacency list for every atom, i.e. a list of atoms to which it is directly bonded and the types of bonds involved. Each term in the list is of the form `KeyAtom-ListOfBondedAtoms` where the `KeyAtom` is of the form `Element(AtomNumber, AtomicWeight, ChargeChirality)` and `ChargeChirality` is of the form `(ApproximateCharge, ChiralityStatusOfKeyAtom)`. Each atom appears twice in the term form. Once as a `KeyAtom` and once in the `ListOfBondedAtoms`. Atoms in the `ListOfBondedAtoms` are of the form `Element(AtomNumber, BondDirection)BondType`. c(1,12,(0,nonchiral))-c(2,left)... means carbon 1 is bonded on its left to carbon 2 by a sigma bond. Thus the `BondDirection` is always specified as if one were sitting on the `KeyAtom` and looking out. The term-form is constructed to give enough information about the stereochemistry of each atom. It gives information about the chirality of the molecule and the different arrangement of atoms on all sides of each atom.

This information is necessary for the isomer recognition of the molecules in the database. The term form can be written in only one unique way and all different types of config rule will generate one identical term-form. Term form 1 and Term-form 2 are term-forms for ethanol (Figure 16) and purine (Figure 1).

Term-form 1: Ethanol

```
c(1,12,(0,nonchiral))-[c(2,left),h(2,right),o(1,up),h(3,down)],
c(2,12,(0,nonchiral))-[h(4,up),c(1,right),h(5,up),h(6,down)],
h(1,1,(0,nonchiral))-[o(1,nil)],
h(2,1,(0,nonchiral))-[c(1,left)],
h(3,1,(0,nonchiral))-[c(1,up)],
h(4,1,(0,nonchiral))-[c(2,right)],
h(5,1,(0,nonchiral))-[c(2,down)],
h(6,1,(0,nonchiral))-[c(2,up)],
o(1,16,(0,nonchiral))-[h(1,nil),c(1,down)].
```
Term-form 2: ATP
3 Search Heuristics and Algorithms

All of the three structure search methods use different algorithms of their own to retrieve the respective target structures. This section will concentrate on the methods employed in the search methods to perform the searches efficiently.

3.1 Descriptors

Descriptors are certain structural features of each compound in Klotho which gives more detailed information about the compound. They can be categorized as primary descriptors and secondary descriptors. The primary descriptors are used to identify the target compounds more distinctly than the secondary descriptors. The secondary descriptors give more general information than the primary descriptors. These constitute structural features which place the compounds in various categories and are used to narrow down the search space to enable a quicker search. Table 2 shows the secondary descriptors which were used in the search tool. The secondary descriptor element combination is the combination of all elements present in the compound. For e.g. purine has the combination of elements \( \text{C,H,N} \) as carbon, hydrogen and nitrogen are the only elements present in purine. Elements specify what individual elements are present in the compound. Class, places the compound in a particular family of compounds. Examples of such families are sugars, nucleotides, amines, alcohols etc. Atom count gives the total number of atoms present in the compound. A compound can be also classified as aliphatic, cyclic or aromatic depending on the presence of chains, cycles or aromatic rings in the compounds. Ring count gives the total number of the rings in the compound. Bond count gives the total number of bonds present in the compound. Molecular weight of the compound is the sum of the atomic weights of each element on the compound. Thus the secondary descriptors give some detailed information about each compound which helps significantly in identifying potential target compounds during a query. Presence of conjugated rings means the presence of adjacent rings. Presence of mixed rings means the presence of different types of rings.

The secondary descriptors and the molecular formula of all the compounds are pre-computed and stored in the database. Hence during a query, when the user enters certain descriptors, the information doesn’t need to be computed. The pre-computed data is accessed and used during the search process. This cuts down considerable computation time by reducing the search space.
and enables a faster search process. Hence biochemical structure information is used to implement heuristics in the search processes. Heuristics is used to screen the database compounds into a condensed group thus reducing the search space and enabling a faster search process. With the use of each descriptor the compounds are condensed into a smaller group. And this smaller condensed group is subjected to the main query predicate.

### 3.2 Isomer Search Algorithms

**Isomer Search:** Isomer search method retrieves all structural isomers and configurational isomers for the query structure. The various isomer types that can be searched by the isomer search method are shown in Table 3. In addition to isomer searches for the different sub-categories, an user can search for all isomers, structural isomers and configurational isomers. The term-forms store essential sterochemical information about every atom and every bond in each molecule. This information combined with logic complying with the basic definitions of each isomer type has been used to create the algorithm for various isomer search methods. The algorithm used in the structural isomer search method is described in Algorithm 1 (Figure 17) and the one used in configurational isomer search method is described in Algorithm 2 (Figure 18).

The sterochemical information stored in the term-form for each molecule succeeds in gathering enough information about the isomer search. To illustrate with an example a part of the term forms for two epimers are shown in Figure 19. Epimers are a class of diastereoisomers where there

<table>
<thead>
<tr>
<th>Type of Descriptor</th>
<th>Name of Descriptor</th>
<th>Description of Descriptor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Primary</td>
<td>Molecular formula</td>
<td>Number of each distinct atom type</td>
</tr>
<tr>
<td>Primary</td>
<td>Name part</td>
<td>Any part of a compound’s name</td>
</tr>
<tr>
<td>Secondary</td>
<td>Element combination</td>
<td>Combination of elements</td>
</tr>
<tr>
<td>Secondary</td>
<td>Elements</td>
<td>Specific elements</td>
</tr>
<tr>
<td>Secondary</td>
<td>Class</td>
<td>Family the compound belongs to e.g. alcohol</td>
</tr>
<tr>
<td>Secondary</td>
<td>Atom count</td>
<td>The total number of atoms in the compound</td>
</tr>
<tr>
<td>Secondary</td>
<td>Compound Type</td>
<td>Aliphatic, cyclic or aromatic compound type</td>
</tr>
<tr>
<td>Secondary</td>
<td>Ring count</td>
<td>The total number of rings in the compound</td>
</tr>
<tr>
<td>Secondary</td>
<td>Bond count</td>
<td>The total number of bonds in the compound</td>
</tr>
<tr>
<td>Secondary</td>
<td>Molecular weight</td>
<td>The total molecular weight of the compound</td>
</tr>
<tr>
<td>Secondary</td>
<td>Presence of conjugated rings</td>
<td>Presence of conjugated rings in the compound</td>
</tr>
<tr>
<td>Secondary</td>
<td>Presence of mixed rings</td>
<td>Presence of mixed rings in the compound</td>
</tr>
</tbody>
</table>

Table 2: Name and description of primary and secondary descriptors.
for input query structure $q$ and a set of descriptors $D$
for all $c \in C$ retrieve set $C_{D_i}$ with descriptors $D_i$ where $D_i \in D$
if $C_{D_i} \neq \emptyset$ then
  compute formula $F$ for $q$
  retrieve $C_{D_i,F}$ for all $c \in C_{D_i}$ with formula $F_i$ where $F_i = F$
  if $C_{D_i,F} \neq \emptyset$ then
    forall $c \in C_{D_i,F}$ retrieve $C_{D_i,FCl_i}$ where $Cl_i$ is same as class of $q = Cl_q$
    if forall $c \in C_{D_i,F}$ class $Cl_i$ is different than $Cl_q$ then
      retrieve $C_{D_i,FCl_i,Ct_i}$
      all $c \in C_{D_i,FCl_i,Ct_i}$ is a functional isomer of $q$
    end if
  end if
for all $c \in C_{D_i,F}$ retrieve $C_{D_i,FCl_i,H}$ having exchange of H atom with $q$
if $C_{D_i,FCl_i,H} \neq \emptyset$ then
  for all $c \in C_{D_i,FCl_i,H}$ retrieve $C_{D_i,FCl_i,Hd}$ having reverse exchange of double bond with $q$
  if $C_{D_i,FCl_i,Hd} \neq \emptyset$ then
    for all $c \in C_{D_i,FCl_i,Hd}$ retrieve $C_{D_i,FCl_i,HdCl_i}$ with compound type $Ct_q$
    for all $c \in C_{D_i,FCl_i,Hd}$ retrieve $C_{D_i,FCl_i,HdCl_i}$ with compound type not $Ct_q$
    if $C_{D_i,FCl_i,HdCl_i} \neq \emptyset$ then
      all $c \in C_{D_i,FCl_i,HdCl_i}$ is a keto-enol tautomer of $q$
    else if $C_{D_i,FCl_i,HdCl_i} \neq \emptyset$ then
      all $c \in C_{D_i,FCl_i,HdCl_i}$ is a ring-chain tautomer of $q$
    end if
  end if
else if $C_{D_i,FCl_i} \neq \emptyset$ then
  for all $c \in C_{D_i,FCl_i}$ retrieve $C_{D_i,FCl_i,Ct}$ with compound type $Ct_q$
  if $C_{D_i,FCl_i,Ct} \neq \emptyset$ then
    for all $c \in C_{D_i,FCl_i,Ct}$ retrieve $C_{D_i,FCl_i,CtB}$ with backbone different than $q$ after excluding functional group
    if $C_{D_i,FCl_i,CtB} \neq \emptyset$ then
      all $c \in C_{D_i,FCl_i,CtB}$ is a skeletal isomer of $q$
    end if
  end if
else if $C_{D_i,F} = \emptyset$ then
  no structural isomers exist for $q$
end if
else if $C_{D_i} = \emptyset$ then
  no result for $q$
end if

Figure 17: Algorithm 1 for structural isomer search
for input query structure \( q \) and a set of descriptors \( D \)
for all \( c \in C \) retrieve set \( C_{D_i} \) with descriptors \( D_i \) where \( D_i \in D \)
if \( C_{D_i} \neq \emptyset \) then
  compute formula \( F \) for \( q \)
  for all \( c \in C_{D_i} \) retrieve \( C_{D,F} \) with formula \( F_i \) where \( F_i = F \)
  if \( C_{D,F} \neq \emptyset \) then
    for all \( c \in C_{D,F} \) retrieve \( C_{D,F,ECH} \) having chiral atoms
    if \( C_{D,F,ECH} = \emptyset \) then
      for all \( c \in C_{D,F,ECH} \) retrieve \( C_{D,F,ECH,C_s} \) in structure rule
      if \( C_{D,F,ECH,C_s} \neq \emptyset \) then
        all \( c \in C_{D,F,ECH,C_s} \) is a geometric isomer of \( q \)
      else if \( C_{D,F,ECH,C_s} = \emptyset \) then
        no geometrical isomers for \( q \)
    end if
  else if \( C_{D,F,ECH} \neq \emptyset \) then
    for all \( c \in C_{D,F,ECH} \) retrieve \( C_{D,F,ECH,H} \) with horizontal plane of symmetry in structure
    if \( C_{D,F,ECH,H} \neq \emptyset \) then
      all \( c \in C_{D,F,ECH,H} \) is a meso-isomer of \( q \)
    end if
  for all \( c \in C_{D,F,ECH} \) retrieve \( C_{D,F,ECH,M} \) with mirror images of \( q \)
  if \( C_{D,F,ECH,M} \neq \emptyset \) then
    all \( c \in C_{D,F,ECH,M} \) is an enantiomer of \( q \)
  else
    for all \( c \in C_{D,F,ECH} \) retrieve \( C_{D,F,ECH,O} \) with chiral bond orientation different around only one chiral atom
    if \( C_{D,F,ECH,O} \neq \emptyset \) then
      for all \( c \in C_{D,F,ECH,O} \) retrieve \( C_{D,F,ECH,OA} \) with difference at anomeric carbon
      if \( C_{D,F,ECH,OA} \neq \emptyset \) then
        all \( c \in C_{D,F,ECH,OA} \) is an anomer of \( q \)
      else if \( C_{D,F,ECH,OA} = \emptyset \) then
        all \( c \in C_{D,F,ECH,OA} \) is an epimer of \( q \)
      end if
    else if \( C_{D,F,ECH,M} = \emptyset \) then
      all \( c \in C_{D,F,ECH} \) is a diastereomer of \( q \)
    end if
  end if
else if \( C_{D,F} = \emptyset \) then
  no configurational isomers exist for \( q \)
end if
else if \( C_{D_i} = \emptyset \) then
  no result for \( q \)
end if

Figure 18: Algorithm 2 for configurational isomer search
Table 3: Types of isomer searches included in the isomer search method

<table>
<thead>
<tr>
<th>Category</th>
<th>Sub-categories</th>
</tr>
</thead>
<tbody>
<tr>
<td>Structural Isomers</td>
<td>Functional isomers, positional isomers, skeletal isomers, tautomers</td>
</tr>
<tr>
<td>Configurational Isomers</td>
<td>Geometrical isomers, enantiomers, diastereomers, anomers, epimers, meso-isomers</td>
</tr>
</tbody>
</table>

**Term form of 'D-mannose'**

\[c(1,12,(0,nonchiral)) \rightarrow [h(1,nil) \land o(6,nil) \land c(2,down) \land \cdots],\]

\[c(2,12,0,chiral)) \rightarrow [o(1,left) \land h(2,right) \land c(1,up) \land c(3,down) \land \cdots],\]

**Term form of 'D-glucose'**

\[c(1,12,(0,nonchiral)) \rightarrow [h(1,nil) \land o(6,nil) \land c(2,down) \land \cdots],\]

\[c(2,12,0,chiral)) \rightarrow [h(2,left) \land o(1,right) \land c(1,up) \land c(3,down) \land \cdots],\]

\[c(3,12,(0,chiral)) \rightarrow [o(2,left) \land h(4,right) \land c(2,up) \land c(4,down) \land \cdots].\]

Figure 19: Term form information for D-mannose and D-glucose for isomer differentiation

is a difference of orientation of bonded groups in only one asymmetric center like D-glucose and D-mannose. The tool first finds those compounds which has orientation different in only one chiral atom which is carbon atom 2 for this case. The orientation of oxygen and hydrogen atom attached to carbon atom 2, in the above epimers are different as shown in the term-form section of interest. In D-mannose the -OH group is on the left and H atom is on the right. On the other hand in D-glucose the exact opposite orientation is observed. This distinguishing feature in the term forms of both the compounds enable to classify them as Epimers. Similarly all the other types of isomers are retrieved by logic which complies with the basic definition of each isomer type.

### 3.3 Substructure Search Algorithm

**Substructure search:** Substructure search method uses subgraph isomorphism to match the query structure with target structures. Pre-computed descriptors are used to narrow down the search
for input query structure $q$ and a set of descriptors $D$
∀$c \in C$ retrieve set $C_{D_i}$ with descriptors $D_i$ where $D_i \in D$
if $C_{D_i} \neq \emptyset$ then
  compute formula $F_q$ for $q$
  retrieve $C_{D_i,F_q}$ for all $c \in C_{D_i}$ with formula $F_i$ where $F_q \subseteq F_i$
  if $C_{D_i,F_q} \neq \emptyset$ then
    for all $c \in C_{D_i,F_q}$ perform atom by atom matching with $q$
    retrieve $S_q$ with all matched compounds
    if $S_q \neq \emptyset$ then
      $S_q$ is the set of compound with $q$ as substructure
    else if $S_q = \emptyset$ then
      no substructure match for $q$
    end if
  else if $C_{D_i,F_q} = \emptyset$ then
    no substructure match for $q$
  end if
else if $C_{i,D} = \emptyset$ then
  no substructure match for $q$
end if

Figure 20: Algorithm 3 for substructure search

space. In the next step the search space is further narrowed down by selecting those structures whose formula subsumes the formula of the query structure. In other words, the number of each distinct atom type in a target structure should be greater than the number of same atom type in the query structure. This step is followed by an atom by atom matching method to match the query structure with screened target structures. The pseudocode for substructure search method is shown in Algorithm 3 (Figure 20).

Figure 21 shows how amine as a substructure is matched in the structure ethylamide. For a substructure-search to succeed, the term form for the substructure needs to be present wholly in the term-form for the target structure. As shown in the figure, the term form for amine is present wholly in the term-form for ethylamide. The atom numbers can be different as they are two separate molecules. But the rest of the information including atom type, bond type, adjacency list should match with each other.

3.4 Isoconnected Structure Search Algorithm

Isoconnected Structure Search: Isoconnected structure search method uses graph isomorphism to match the query structure with target structures. This method enables the user to retrieve an
Figure 21: Term form information for amine and ethylamide for substructure search

Term form of Amine, \([n-1,h-2]\)

\[\text{n}(1,14,0,\text{nonchiral})\rightarrow [\text{n}(1,\text{nil})\rightarrow ,\text{n}(2,\text{nil})\rightarrow ,]\]

\[\text{h}(1,1,0,\text{nonchiral})\rightarrow [\text{n}(1,\text{nil})\rightarrow ]\]

\[\text{h}(2,1,0,\text{nonchiral})\rightarrow [\text{n}(1,\text{nil})\rightarrow ]\]

Term form of Ethylamide, \([c\cdot 3, n-1, o\cdot 1, h-6]\)

\[\text{c}(2,12,0,\text{nonchiral})\rightarrow [\text{c}(1,\text{left})\rightarrow ,\text{c}(3,\text{right})\rightarrow ,\text{h}(1,\text{up})\rightarrow ,\text{h}(2,\text{down})\rightarrow ]\]

\[\text{c}(1,12,0,\text{nonchiral})\rightarrow [\text{h}(3,\text{nil})\rightarrow ,\text{c}(2,\text{right})\rightarrow ,\text{h}(4,\text{nil})\rightarrow ]\]

\[\text{c}(3,12,0,\text{nonchiral})\rightarrow [\text{c}(2,\text{left})\rightarrow ,\text{o}(1,\text{nil})\rightarrow ,\text{n}(1,\text{nil})\rightarrow ]\]

\[\text{n}(1,14,0,\text{nonchiral})\rightarrow [\text{h}(5,\text{nil})\rightarrow ,\text{h}(6,\text{nil})\rightarrow ,\text{c}(3,\text{nil})\rightarrow ]\]

\[\text{h}(1,1,0,\text{nonchiral})\rightarrow [\text{c}(2,\text{down})\rightarrow ]\]

\[\text{h}(2,1,0,\text{nonchiral})\rightarrow [\text{c}(2,\text{up})\rightarrow ]\]

\[\text{h}(3,1,0,\text{nonchiral})\rightarrow [\text{c}(1,\text{nil})\rightarrow ]\]

\[\text{h}(4,1,0,\text{nonchiral})\rightarrow [\text{c}(1,\text{nil})\rightarrow ]\]

\[\text{o}(1,16,0,\text{nonchiral})\rightarrow [\text{c}(3,\text{nil})\rightarrow ]\]

\[\text{h}(5,1,0,\text{nonchiral})\rightarrow [\text{n}(1,\text{nil})\rightarrow ]\]

\[\text{h}(6,1,0,\text{nonchiral})\rightarrow [\text{n}(1,\text{nil})\rightarrow ]\]
for input query structure $q$ and a set of descriptors $D$
\forall c \in C$ retrieve set $C_D$ with descriptors $D_i$ where $D_i \in D$
\textbf{if} $C_D \neq \emptyset$ \textbf{then}
\hspace{1em} compute formula $F$ for $q$
\hspace{1em} retrieve $C_{D,F}$ for all $c \in C_D$ with formula $F_i$ where $F_i = F$
\hspace{1em} \textbf{if} $C_{D,F} \neq \emptyset$ \textbf{then}
\hspace{2em} for all $c \in C_{D,F}$ perform atom by atom matching with $q$
\hspace{2em} retrieve $C_{D,F_I}$ with all matched compounds
\hspace{2em} \textbf{if} $C_{D,F_I} \neq \emptyset$ \textbf{then}
\hspace{3em} $C_{D,F_I}$ is the set of compound with structure exactly same as $q$
\hspace{3em} \textbf{if} $C_{D,F_I} \neq \emptyset$ \textbf{then}
\hspace{4em} for all $c \in C_{D,F_I}$ retrieve set $C_{D,F_IEX}$ with stereochemistry exactly same as $q$
\hspace{2em} \textbf{else if} $C_{D,F_I} = \emptyset$ \textbf{then}
\hspace{3em} no isoconnected structure match for $q$
\hspace{1em} \textbf{end if}
\hspace{1em} \textbf{else if} $S_q = \emptyset$ \textbf{then}
\hspace{2em} no isoconnected structure match for $q$
\hspace{1em} \textbf{end if}
\hspace{1em} \textbf{else if} $C_D = \emptyset$ \textbf{then}
\hspace{2em} no isoconnected structure match for $q$
\hspace{1em} \textbf{end if}
\hspace{1em} \textbf{else if} $C_{i,D} = \emptyset$ \textbf{then}
\hspace{2em} no isoconnected structure match for $q$
\hspace{1em} \textbf{end if}

Figure 22: Algorithm 4 for isoconnected structure search

exact structure with exact stereochemistry as the query compound. It also has option to retrieve structures that are isoconnected with the query structure. This method also uses the pre-computed descriptor information to narrow down the search space. The next screening criteria for the search is that the number of each distinct atom type in the target compound should be exactly equal to the same of the query compound. This step is followed by an atom by atom matching method to match query structure with screened target structures. The psuedocode for isoconnected structure search method is shown in Algorithm 4 (Figure 22).

To match two compounds after initial screening and heuristics, atom by atom matching algorithm was used. This algorithm matches keypairs of atoms and progresses through the adjacency list of the matched atoms. This algorithm is described in Algorithm 5 (Figure 23).

A test suite was prepared to test the structure search tool. The queries in the search tool were tested in this test suite with all the compounds in our database Klotho [14]. The queries were categorized as isomer search queries, substructure search queries, isoconnected structure search
for term form $TQ$ for query structure $q$
for term form $TT$ for target structure $t$
for all $t \in TQ$ retrieve $T_{Ch'}$ after removing atom charges for $q$
for all $t \in TQ_{Ch'}$ retrieve $T_{Ch,N'}$ after removing atom numbers for $q$
for all $t \in TQ_{Ch,N'}$ retrieve $T_{Ch,N',D'}$ after removing atom directions for $q$
for all $t \in TQ_{Ch,N',D'}$ pick key-atoms $KA_q$ in $q$
pick key-atoms $KA_T$ in target structure
for all $k \in KA_q$ find least occurring atom type $L_q$
for all $t \in T_{Ch,N',D'}$ pick key pair $KPL_q$ with $L_q$
find $L_q$ in $KA_T$
check if LOAT key pair present in target structure
\begin{verbatim}
if $KPL_q \in TT$ then
    retrieve $RKPL_q$ by reversing bond direction of $KPL_q$
    if $RKPL_q \in TT$ then
        repeat same with the adjacency list of $KPL_q$
    else if $RKPL_q \notin TT$ then
        no atom by atom match
    end if
else if $KPL_q \notin TT$ then
    no atom by atom match
end if
\end{verbatim}

Figure 23: Algorithm 5 for atom by atom matching

queries and general structure search queries. The purpose of creating the test suite was to test the queries with both +ve and -ve controls. Firstly the categorical search methods should retrieve all compounds the specific methods are supposed to retrieve. And secondly the methods should not retrieve any compounds they are not supposed to retrieve. Specific examples of the test suite queries are given later in the Results section.
4 Validation and Benchmarking of Search Procedures

<table>
<thead>
<tr>
<th>Compound Type</th>
<th>Number of Compounds</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aliphatic</td>
<td>1243</td>
</tr>
<tr>
<td>Aliphatic-Cyclic</td>
<td>913</td>
</tr>
<tr>
<td>Aromatic</td>
<td>1345</td>
</tr>
</tbody>
</table>

Table 4: Compound type distribution in Klotho database

Table 4 gives the distribution of compound types in Klotho database. Having the descriptor information pre-computed, decreased the search time during an actual query. The pre-computed data for each molecule was stored as lists in files. When query was submitted the indexed descriptor information was referred to in the respective file and only those compounds matching the descriptor information underwent the matching process with the query compound. Figure 24 is a flow diagram depicting the use of precomputed descriptor information in the search methods.

4.1 Isomer Search

The chemical definition of structural and configurational isomers was used to conduct the isomer searches. Individual flow diagrams for each of the isomer search process show the individual algorithms more closely. The queries related to isomer search are listed below in Table 5 which contains all the important queries.

**Functional Isomers:** The database was initially screened to collect all the compounds with same formula. By definition, functional isomers have different functional groups. The secondary descriptor, class, gives information about the various functional groups present in a compound. Hence after the initial screening, all screened compounds are checked to see what class they belong to by referring to the pre-computed values. Also another screening is done to retrieve only those compounds that are same type of molecule as the query compound, i.e. both are rings or both are chains. Those target compounds that belong to different class and are same type of molecule as the query compound are the functional isomers. Fig 25 shows the flow diagram for the search.
<table>
<thead>
<tr>
<th>No.</th>
<th>Query</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Find all isomers for query compound</td>
</tr>
<tr>
<td>2</td>
<td>Find all structural isomers for query compound</td>
</tr>
<tr>
<td>3</td>
<td>Find all configurational isomers for query compound</td>
</tr>
<tr>
<td>4</td>
<td>Find all positional isomers for query compound with descriptor combinations</td>
</tr>
<tr>
<td>5</td>
<td>Find all skeletal isomers for query compound with descriptor combinations</td>
</tr>
<tr>
<td>6</td>
<td>Find all functional isomers for query compound with descriptor combinations</td>
</tr>
<tr>
<td>7</td>
<td>Find all tautomers for query compound with descriptor combinations</td>
</tr>
<tr>
<td>8</td>
<td>Find all geometric isomers for query compound with descriptor combinations</td>
</tr>
<tr>
<td>9</td>
<td>Find all enantiomers for query compound with descriptor combinations</td>
</tr>
<tr>
<td>10</td>
<td>Find all diastereomers for query compound with descriptor combinations</td>
</tr>
<tr>
<td>11</td>
<td>Find all epimers for query compound with descriptor combinations</td>
</tr>
<tr>
<td>12</td>
<td>Find all anomers for query compound with descriptor combinations</td>
</tr>
<tr>
<td>13</td>
<td>Find all meso-isomers for query compound with descriptor combinations</td>
</tr>
<tr>
<td>14</td>
<td>Find all positional isomers for query compound in a list of compounds</td>
</tr>
<tr>
<td>15</td>
<td>Find all skeletal isomers for query compound in a list of compounds</td>
</tr>
<tr>
<td>16</td>
<td>Find all functional isomers for query compound in a list of compounds</td>
</tr>
<tr>
<td>17</td>
<td>Find all tautomers for query compound in a list of compounds</td>
</tr>
<tr>
<td>18</td>
<td>Find all geometric isomers for query compound in a list of compounds</td>
</tr>
<tr>
<td>19</td>
<td>Find all enantiomers for query compound in a list of compounds</td>
</tr>
<tr>
<td>20</td>
<td>Find all diastereomers for query compound in a list of compounds</td>
</tr>
<tr>
<td>21</td>
<td>Find all epimers for query compound in a list of compounds</td>
</tr>
<tr>
<td>22</td>
<td>Find all anomers for query compound in a list of compounds</td>
</tr>
<tr>
<td>23</td>
<td>Find all meso-isomers for query compound in a list of compounds</td>
</tr>
<tr>
<td>24</td>
<td>Find all structures with varied descriptor combinations and multiple substructures</td>
</tr>
<tr>
<td>25</td>
<td>Find all structures with q as a substructure in a list of compounds</td>
</tr>
<tr>
<td>26</td>
<td>Find all structures having isooconnected structure as q</td>
</tr>
<tr>
<td>27</td>
<td>Find all structures with q as in a list of compounds</td>
</tr>
<tr>
<td>28</td>
<td>Find all structures with q with descriptor values</td>
</tr>
<tr>
<td>29</td>
<td>Find all descriptor values of compound q</td>
</tr>
</tbody>
</table>

Table 5: Main query table. These queries are implemented in the front end.
Figure 24: Flow diagram showing the use of pre-computed descriptor information in various structure search methods.

Figure 25: Flow chart for functional isomer search
Positional Isomers: Positional isomers are isomers with same backbone and same functional group at different positions in the skeleton of the compound. So after initial screening for database compounds with the same formula, the pre-computed data is checked for compounds with same class as query compound. Since positional isomers have same functional group, they will belong to the same class. Like the functional isomers these isomers also need to be the same type of molecule as the query compound. Fig 26 shows the flow diagram for the search.

Skeletal isomers: The skeletal isomers basically have different skeleton or back bones. They have the same functional group so they belong to the same class of compounds. Also they need to belong to the same type of molecule group. So in search for skeletal isomers, the compounds with
same formula were screened according to these conditions. As a final check the functional group is removed from both query and target compounds. If the resulting structures are not identical, then they are skeletal isomers as for skeletal isomers the backbone should be different. Figure 27 shows the flow diagram for the search.

**Tautomers**: Tautomerization mainly deals with keto-enol conversion in which there is exchange of double bonds and H atoms. Since a keto group is changed to an alcohol functional group, the tautomers will belong to different class. Also ring chain tautomers will belong to different types of molecule while keto-enol and other types of isomers will belong to same type of molecule. Once the screenings for formula, class and molecule type are is done, the H atoms and multiple bonds are
Figure 28: Flow chart for tautomer search

removed from both query and target structure. If the skeleton of the resulting structures match, then they are tested for keto-enol tautomers. If one skeleton is substructure of the other then they are tested for ring-chain tautomers. The number of H atoms attached to every non-H atom is counted in both the structures. A check is made to see which atoms have different H atoms in both query and target structures. Those two atoms are subjected to another check to see if there was an exchange of double bond with them. Target compounds that pass all the checks are tautomers. Figure 28 shows the flow diagram of the search.

For configurational isomers along with the initial screening for compounds with same formula, two other screenings are also done. Those compounds with same molecular type and same class are also screened out and configurational isomer search was done only on these compounds. This overall
screening basically retrieved those compounds whose full structure is identical to the full structure of the query compound. Thus the use of heuristic here has considerably reduced the search space and speeded up the search process.

Geometric Isomers: Geometric isomers differ in arrangement around multiple bonds. In term-form, the cis and trans isomers are defined accordingly with the terms cis or trans. So after initial screening of compounds with identical structure first those compounds are retrieved which have the term cis- or trans- in their config rule. Then from these reduced result, compounds with atoms having the cis or trans term opposite to the same in query compound are selected as final results. Figure 29 shows a flow chart for geometric isomer search.
Diastereomers: After the initial screening for identical structures as the query compound, only those compounds which have chiral centers are screened. Then the term forms for these compounds are checked to retrieve those compounds in which all chiral center configuration is not mirror image i.e. not opposite of all chiral center configurations in query compound. So there should be at least one chiral center with different configuration. These short listed compounds are then the diastereomers of the query compound. Another search is conducted in the diastereomers to find compounds which have different configuration in exactly one chiral center. Out of these if that chiral center is the anomeric carbon, then the diastereomers are termed anomers and the rest are termed epimers. Figure 30 shows a flow chart for diastereomer search.
Enantiomers: In the enantiomer search process, after the initial screening for identical structures as the query compound, only those compounds with chiral centers are screened out. Then the term forms for these compounds are checked to retrieve only those compounds in which all chiral center configuration is the mirror image, i.e. exactly the opposite, of all chiral center configurations in query compound. Figure 31 shows a flow chart for diastereomer search.

Meso-isomers: In the meso-isomer search process, after the initial screening for identical structures as the query compound, compounds with chiral centers are first screened. Then the term forms for these compounds are checked to retrieve only those compounds in which there is even number of chiral centers. Since meso-isomers have a plane of symmetry through the middle of the compound,
it is essential to have an even number of chiral centers. Then all chiral center configurations of the upper half of the molecule are checked to see if they are identical to the same of the lower half of the molecule. The compounds that pass this check are the meso-isomers of the query compound. Figure 33 shows a flow chart for meso-isomer search.

Benchmarked data is shown in Table 6 for all the isomer searches. It contains data for different types of isomer searches for different compounds. These queries were run against all compounds present in our database Klotho [14]. The CPU runtime and memory used varies depending on the isomer search type and the query structure $q$. The number of possible compounds (fifth column in Table 6) refers to those compounds which have same molecular formula as the query structure $q$. 

Figure 32: Flow chart for meso-isomers search
### Table 6: Isomer search data.

<table>
<thead>
<tr>
<th>Isomer Type</th>
<th>Query Compound</th>
<th>CPU Runtime (s)</th>
<th>Total Memory (MB)</th>
<th>Effective Search Space</th>
<th>No of Hits</th>
</tr>
</thead>
<tbody>
<tr>
<td>Functional</td>
<td>acetone</td>
<td>0.39</td>
<td>4.1311</td>
<td>8</td>
<td>6</td>
</tr>
<tr>
<td>Positional</td>
<td>D-3-hydroxyproline</td>
<td>0.97</td>
<td>4.3942</td>
<td>7</td>
<td>4</td>
</tr>
<tr>
<td>Skeletal</td>
<td>trimethylamine</td>
<td>0.33</td>
<td>3.0923</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>Tautomer</td>
<td>4-pyridinone</td>
<td>0.18</td>
<td>0.4656</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>Geometric</td>
<td>(2R,4E)-1-alkenyl-2-acylglycerol</td>
<td>0.15</td>
<td>1.2588</td>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>Anomer</td>
<td>alpha-D-lyxopyranose</td>
<td>0.12</td>
<td>2.2348</td>
<td>15</td>
<td>1</td>
</tr>
<tr>
<td>Epimer</td>
<td>(1S,2R)-1-hydroxyalkyl-sn-glycerol</td>
<td>0.09</td>
<td>1.2569</td>
<td>7</td>
<td>1</td>
</tr>
<tr>
<td>Diastereomer</td>
<td>(2R,3R,7R,11R)-2-hydroxyphytanoyl</td>
<td>2.05</td>
<td>2.1964</td>
<td>16</td>
<td>11</td>
</tr>
<tr>
<td>Enantiomer</td>
<td>D-mannose</td>
<td>0.56</td>
<td>3.6737</td>
<td>14</td>
<td>1</td>
</tr>
<tr>
<td>Meso-isomer</td>
<td>2S,3R-1,2,3,4-Butanetetrol</td>
<td>0.06</td>
<td>1.8424</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

#### 4.2 Substructure Search

The substructure algorithm was able to successfully and exhaustively retrieve all the substructures. After the initial screening of those compounds which matched the secondary descriptor inputs, the compounds whose formula subsumes the query formula were computed from them. The short listed compounds were then subjected to atom by atom matching to check the presence of the query substructure in the target compounds. Fig 33 is a flow chart of the substructure search algorithm which was used in the search tool. Besides the main queries shown in Table 6 a number of small queries were also developed which are shown in Table 7 below:

Tables 8 and 9 show a comparison of the cpu runtimes for substructure search, with purine, without any descriptor and with different number of descriptors. As evident from the tables and corresponding figures (Figure 34 and Figure 35 respectively) including descriptors in the search methods decreases the processing time. A substructure search query with three or more descriptors performs the search process in less than a second. So an user can search by choosing only three descriptors out of the twelve descriptors available and still save considerable time. This feature is beneficial as many users may not be aware of all descriptor values for a specific compound and need not put all of them. Since the descriptor information is processed and stored the search time was decreased when an user queries the database. The best descriptor according to Table 8 is the molecular formula which narrows down the search space to the smallest groups. The next best ones are atom count, bond count and molecular weight. The descriptors compound type, class, conjugated rings and mixed rings are not a good choice by themselves and should be used in combination with other descriptors.
<table>
<thead>
<tr>
<th>No.</th>
<th>Query</th>
</tr>
</thead>
<tbody>
<tr>
<td>30</td>
<td>Find all compounds with given substructure where input is name of substructure</td>
</tr>
<tr>
<td>31</td>
<td>Find all compounds with given substructure where input is config rule of substructure</td>
</tr>
<tr>
<td>32</td>
<td>Find all compounds with given substructure and name part</td>
</tr>
<tr>
<td>33</td>
<td>Find all compounds with given substructure and class</td>
</tr>
<tr>
<td>34</td>
<td>Find all compounds with given substructure and compound type</td>
</tr>
<tr>
<td>35</td>
<td>Find all compounds with given substructure and number of rings</td>
</tr>
<tr>
<td>36</td>
<td>Find all compounds with given substructure, total atom count, compound type and number of rings</td>
</tr>
<tr>
<td>37</td>
<td>Find all compounds with given substructure and total atom count range</td>
</tr>
<tr>
<td>38</td>
<td>Find all compounds with given substructure and bond count range</td>
</tr>
<tr>
<td>39</td>
<td>Find all compounds with multiple substructures</td>
</tr>
<tr>
<td>40</td>
<td>Find all compounds with given substructure and multiple class</td>
</tr>
<tr>
<td>41</td>
<td>Find all compounds with given substructure and class and compound type</td>
</tr>
<tr>
<td>42</td>
<td>Find all compounds with given substructure and class, compound type and ring count</td>
</tr>
<tr>
<td>43</td>
<td>Find all compounds with given substructure and conjugated rings</td>
</tr>
<tr>
<td>44</td>
<td>Find all compounds with given substructure and mixed rings</td>
</tr>
<tr>
<td>45</td>
<td>Find all compounds with given molecular weight</td>
</tr>
<tr>
<td>46</td>
<td>Find all compounds with given ring count</td>
</tr>
<tr>
<td>47</td>
<td>Find all compounds with given class</td>
</tr>
<tr>
<td>48</td>
<td>Find all compounds with given total atom count</td>
</tr>
<tr>
<td>49</td>
<td>Find all compounds with conjugate rings</td>
</tr>
<tr>
<td>50</td>
<td>Find all compounds with given compound type</td>
</tr>
<tr>
<td>51</td>
<td>Find all compounds with given formula</td>
</tr>
<tr>
<td>52</td>
<td>Find all compounds with config rule</td>
</tr>
<tr>
<td>53</td>
<td>Find all compounds with name part</td>
</tr>
<tr>
<td>54</td>
<td>Find all compounds with given multiple class</td>
</tr>
<tr>
<td>55</td>
<td>Find all compounds with name part and compound type</td>
</tr>
<tr>
<td>56</td>
<td>Find all compounds with given name part and formula</td>
</tr>
<tr>
<td>57</td>
<td>Find all compounds with given name part and total atom count</td>
</tr>
<tr>
<td>58</td>
<td>Find all compounds with name part and class</td>
</tr>
<tr>
<td>59</td>
<td>Find all compounds with given total atom count and formula</td>
</tr>
<tr>
<td>60</td>
<td>Find all compounds with given class, atom count, compound type, ring count</td>
</tr>
<tr>
<td>61</td>
<td>Find all compounds with given class, atom count, compound type, ring count, bond count</td>
</tr>
<tr>
<td>62</td>
<td>Find all compounds with mixed rings</td>
</tr>
<tr>
<td>63</td>
<td>Find all compounds with total atom count range</td>
</tr>
<tr>
<td>64</td>
<td>Find all compounds with bond count range</td>
</tr>
</tbody>
</table>

Table 7: Additional structure search queries
Figure 33: Flow chart for substructure search

<table>
<thead>
<tr>
<th>Descriptors in Query</th>
<th>CPU Runtime (s)</th>
<th>No. Of compounds retrieved</th>
</tr>
</thead>
<tbody>
<tr>
<td>none</td>
<td>12.1</td>
<td>302</td>
</tr>
<tr>
<td>element combination</td>
<td>6.02</td>
<td>156</td>
</tr>
<tr>
<td>specific elements</td>
<td>8.3</td>
<td>156</td>
</tr>
<tr>
<td>molecular formula</td>
<td>0.73</td>
<td>16</td>
</tr>
<tr>
<td>name part</td>
<td>8.68</td>
<td>55</td>
</tr>
<tr>
<td>atom count</td>
<td>1.94</td>
<td>38</td>
</tr>
<tr>
<td>bond count</td>
<td>1.93</td>
<td>38</td>
</tr>
<tr>
<td>molecular weight</td>
<td>2.06</td>
<td>42</td>
</tr>
<tr>
<td>ring count</td>
<td>8.21</td>
<td>197</td>
</tr>
<tr>
<td>compound type</td>
<td>17.71</td>
<td>302</td>
</tr>
<tr>
<td>class</td>
<td>24.57</td>
<td>270</td>
</tr>
<tr>
<td>conjugated rings</td>
<td>11.36</td>
<td>302</td>
</tr>
<tr>
<td>mixed rings</td>
<td>15.31</td>
<td>267</td>
</tr>
</tbody>
</table>

Table 8: Comparison of CPU runtime for substructure search of purine (C5H4N4) without and with one descriptor.
Figure 34: Plot for comparison of different descriptors. Data taken from Table 8

<table>
<thead>
<tr>
<th>Descriptors in Query</th>
<th>CPU Runtime (s)</th>
<th>No. of compounds retrieved</th>
</tr>
</thead>
<tbody>
<tr>
<td>none</td>
<td>12.1</td>
<td>302</td>
</tr>
<tr>
<td>class</td>
<td>24.57</td>
<td>270</td>
</tr>
<tr>
<td>name part</td>
<td>8.68</td>
<td>55</td>
</tr>
<tr>
<td>element combination</td>
<td>6.02</td>
<td>156</td>
</tr>
<tr>
<td>atom count</td>
<td>1.94</td>
<td>38</td>
</tr>
<tr>
<td>molecular formula</td>
<td>0.73</td>
<td>16</td>
</tr>
<tr>
<td>name part, class</td>
<td>7.5</td>
<td>54</td>
</tr>
<tr>
<td>element combination, class</td>
<td>6.07</td>
<td>156</td>
</tr>
<tr>
<td>atom count, class</td>
<td>1.92</td>
<td>38</td>
</tr>
<tr>
<td>atom count, element combination</td>
<td>1.76</td>
<td>38</td>
</tr>
<tr>
<td>name part, element combination</td>
<td>1.57</td>
<td>38</td>
</tr>
<tr>
<td>name part, atom count</td>
<td>0.76</td>
<td>16</td>
</tr>
<tr>
<td>element combination, formula</td>
<td>0.76</td>
<td>16</td>
</tr>
<tr>
<td>element combination, atom count, class</td>
<td>1.82</td>
<td>38</td>
</tr>
<tr>
<td>element combination, name part, class</td>
<td>1.61</td>
<td>38</td>
</tr>
<tr>
<td>name part, atom count, class</td>
<td>0.8</td>
<td>16</td>
</tr>
<tr>
<td>element combination, atom count, formula</td>
<td>0.76</td>
<td>16</td>
</tr>
<tr>
<td>element combination, name, formula</td>
<td>0.76</td>
<td>16</td>
</tr>
<tr>
<td>name part, formula, class</td>
<td>0.76</td>
<td>16</td>
</tr>
<tr>
<td>element combination, name, atom count</td>
<td>0.76</td>
<td>16</td>
</tr>
<tr>
<td>name part, formula, class</td>
<td>0.75</td>
<td>16</td>
</tr>
<tr>
<td>name part, formula, atom count</td>
<td>0.75</td>
<td>16</td>
</tr>
<tr>
<td>element combination, formula, class, name part</td>
<td>0.76</td>
<td>16</td>
</tr>
<tr>
<td>element combination, name part, atom count, class</td>
<td>0.76</td>
<td>16</td>
</tr>
<tr>
<td>name part, formula, atom count, class</td>
<td>0.76</td>
<td>16</td>
</tr>
</tbody>
</table>

Table 9: Comparison of CPU runtime for substructure search of purine (C5H4N4) with different combinations of descriptors.
4.3 Isoconnected Structure Search

The isoconnected structure algorithm was able to successfully and exhaustively retrieve all the substructures. After the initial screening of those compounds with same molecule type and same class, those with formula equal to query compound those compounds were screened out. These compounds were then subjected to atom by atom matching process of the term-form to retrieve the compounds with isoconnected structure match. The flow chart for the isoconnectivity chart is shown below in Figure 36.

4.4 General Queries

Some general queries were also created which are not confined to isomer, substructure or exact structure search methods. The general queries can be used to search for compounds with specific descriptor values. These queries retrieve pre-computed descriptor information. These general queries are also listed both in Table 6 and Table 7.
Figure 36: Flow chart for the isoconnectivity search method
4.5 Verification of Results

The results obtained from queries were verified to ensure correctness of the search methods. All three structure search method results were verified by different test queries. For the isomer search methods, since the number of retrieved structures was not big, manual checking was possible. The retrieved structures were checked to see if they complied with the definition of various types of isomers. Also compounds were checked against a list of non-isomers to ensure correctness of the query. For substructure search process various strategic queries were tested. For e.g. purine was tested against a list of lipids to check if any structures were retrieved or not. For isoconnected structure search queries the number of compounds retrieved was not big so manual checking was possible. Table 10 and Table 11 show some of the verification queries used for the different search methods. Figure 37 shows an example of verification for isomer search while Figure 36 shows an example of verification for substructure search.
<table>
<thead>
<tr>
<th>Test Query</th>
<th>Compound set</th>
<th>True +ve</th>
<th>True -ive</th>
<th>False +ive</th>
<th>False -ive</th>
<th>Verification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Geometric Isomers 1</td>
<td>(2R,3R,4E)-2-(2-hydroxyacyl)sphingosine, (2R,3S,4E)-2-(2-hydroxyacyl)sphingosine</td>
<td>Z orientation</td>
<td>E-orientation</td>
<td>None</td>
<td>None</td>
<td>E orientation is not geometric isomer</td>
</tr>
<tr>
<td>Geometric Isomers 2</td>
<td>maleate,maleyl,</td>
<td>maleate</td>
<td>maleyl</td>
<td>None</td>
<td>None</td>
<td>maleyl is not geometric isomer</td>
</tr>
<tr>
<td>E-nantioners 2</td>
<td>L-mannose,L-gulose</td>
<td>L-mannose</td>
<td>L-gulose</td>
<td>None</td>
<td>None</td>
<td>L-gulose is an epimer</td>
</tr>
<tr>
<td>Anomers 1</td>
<td>beta-D-xlyopyranose,</td>
<td>beta-D-xlyopyranose</td>
<td>alpha-L-xlyopyranose</td>
<td>None</td>
<td>None</td>
<td>alpha-L cpd not anomer</td>
</tr>
<tr>
<td>(2S,5R)-2,2”-iminodipropionic acid</td>
<td>(2S,5R)-2,2”-iminodipropionic acid, (2R,SS)-2,2”-iminodipropionic acid, (2R,SS)-2,2”-iminodipropionic acid</td>
<td>(2S,5R)-2,2”-iminodipropionic acid</td>
<td>(2R,SS)-2,2”-iminodipropionic acid</td>
<td>None</td>
<td>None</td>
<td></td>
</tr>
<tr>
<td>Positional isomers 1</td>
<td>1,3-dimethylbenzene,1,4-dimethylbenzene, 1,2-dimethylbenzene, m-xylene,p-xylene</td>
<td>1,3-dimethylbenzene,1,4-dimethylbenzene, m-xylene,p-xylene</td>
<td>1,2-dimethylbenzene,</td>
<td>None</td>
<td>None</td>
<td>1,2-dimethylbenzene same as o-xylene</td>
</tr>
<tr>
<td>Positional isomers 2</td>
<td>1,4-alpha-D-galacturionate-4-yl, 2-keto-L-gulono-1,4-lactone</td>
<td>1,4-alpha-D-galacturionate-4-yl</td>
<td>2-keto-L-gulono-1,4-lactone</td>
<td>None</td>
<td>None</td>
<td>2-keto-L-gulono-1,4-lactone is not positional isomer</td>
</tr>
<tr>
<td>Skeletal isomers 1</td>
<td>2-aminopropane,CH3C(NH2)HCH3, N-methylethylamin-1-yl,propylamine</td>
<td>2-aminopropane,CH3C(NH2)HCH3</td>
<td>N-methylethylamin-1-yl,propylamine</td>
<td>None</td>
<td>None</td>
<td>N-methylethylamin-1-yl is not a skeletal isomer</td>
</tr>
<tr>
<td>Functional isomers 1</td>
<td>1-propen-1-ol,1-propen-3-ol, propionyl,propanone-3-yl</td>
<td>1-propen-1-ol,1-propen-3-ol</td>
<td>propionyl,propanone-3-yl</td>
<td>None</td>
<td>None</td>
<td>propionyl,propanone-3-yl are not a functionalal isomers</td>
</tr>
<tr>
<td>Test Query</td>
<td>Compound set</td>
<td>True +ve</td>
<td>True -ive</td>
<td>False +ive</td>
<td>False -ive</td>
<td>Verification</td>
</tr>
<tr>
<td>------------------------------------------------</td>
<td>-------------------------------------</td>
<td>----------</td>
<td>-----------------</td>
<td>------------</td>
<td>------------</td>
<td>------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Substructure search with purine</td>
<td>behenyl,sphingosine</td>
<td>None</td>
<td>behenyl,sphingosine</td>
<td>None</td>
<td>None</td>
<td>Purine is not a substructure of lipids</td>
</tr>
<tr>
<td>Substructure Search with stearyl</td>
<td>behenyl,sphingosine</td>
<td>behenyl,arachidyl</td>
<td>sphingosine</td>
<td>None</td>
<td>None</td>
<td>sphingosine does not contain stearyl</td>
</tr>
<tr>
<td>Substructure Search with pyrimidine</td>
<td>Compounds with atom count between 38 and 39</td>
<td>All compounds with pyrimidine substructure</td>
<td>Compounds without purimidine as substructure</td>
<td>None</td>
<td>urate,D-ribonucleotide</td>
<td>Query worked for all but one compound</td>
</tr>
<tr>
<td>Search compounds with name part 'CoA' and aromatic type; Substructure search with with coenzymeA; Find difference</td>
<td>All compounds in Klotho database</td>
<td>'1-enzymeA-yl','enzymeA, 'enzymeA-yl'</td>
<td>dephospho-CoA</td>
<td>dephospho-CoA contains a phosphorus atom less than coenzymeA</td>
<td>The three compounds have different names</td>
<td></td>
</tr>
<tr>
<td>Exact structure search for D-glucose</td>
<td>D-altrose,D-fructose</td>
<td>D-altrose retrieved</td>
<td>D-fructose</td>
<td>None</td>
<td>None</td>
<td>D-fructose has different structure</td>
</tr>
<tr>
<td>Exact structure search with D-glucose for aliphatic cpds Exact structure search with D-glucose cpds with no rings</td>
<td>All compounds in database</td>
<td>Both queries gave same result of glucose</td>
<td>All aliphatic compounds with no rings and not exact structure</td>
<td>None</td>
<td>None</td>
<td>D-fructose has different structure</td>
</tr>
</tbody>
</table>

Table 11: Verification of some substructure search and exact structure search queries
4.6 User Interface

The user interface for the structure search was kept simple and user friendly. The format of the forms were kept in compliance with the other search methods in the Klotho system. Figure 39 shows the front page of the entire structure search system. Figure 40 is a screenshot of the substructure search page. The individual search method pages show examples of what kind of data to input. An example screen-shot is shown in Figure 41. This screenshot is for the query for substructure search with the query substructure and descriptors class, compound type and ring count. For this particular query, the user inputs the substructure name, and the class, compound type and ring count for the target structures. Figure 42 shows a sample output page for a query for all structural isomers for the compound D-3-hydroxyproline. The output includes the input information and the name of the query. The structures retrieved are arranged in a separate columns for each isomer type. Each structure retrieved is also a link to the compound page in Klotho database if available. When the page is not available the name of the compound is output without a link. Fig 43 is another example of an output page where the query is for all compounds containing name part 'CoA' and substructure coenzymeA.
Figure 39: Screenshot of user interface of main structure search page

Figure 40: Screenshot of user interface of substructure search page

Figure 41: Screenshot of user interface of functional isomer search form
Your Query: find all structural isomers of query compound, used the following data:

- Compound name: D-3-hydroxyproline
- Priority status: public
- Data source: ChEBI

Query results:

<table>
<thead>
<tr>
<th>Functional Isomers</th>
<th>Positional Isomers</th>
<th>Totamers</th>
<th>Molecular Isomers</th>
</tr>
</thead>
<tbody>
<tr>
<td>No functional isomer</td>
<td>0-4-hydroxyproline</td>
<td>No isomers</td>
<td>No steric isomer</td>
</tr>
<tr>
<td></td>
<td>0-4-hydroxyproline</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0-4,6-dehydroxyproline</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 42: Screenshot of result for structural isomers search

Your Query: find all compounds with the descriptors in your query and a list of substructures, used the following data:

- List of compounds checked: coenzymeA
- Element combination: chains
- Elements present: none
- Formula: none, Name Part: CoA, Atom count > x: none, Atom count < x: none, Bond count > x: none, Bond count < x: none, Molecular weight > x: none, Molecular weight < x: none, Number of rings: none
- Compound class: aromatic
- Compound class: none
- Conjugated rings present: yes
- Hetero rings present: none
- Priority status: public
- Data source: mendei

Query results:

| (1)-hydroxypropane-1-carboxyl-CoA |
| (2)-diaminobutyrate-CoA |
| (2R,3S)-2-hydroxy-2-methylbutyr-CoA |
| (2R,3S,4S)-2-hydroxypropane-1-carboxyl-CoA |
| (2R,3S,4S)-2-hydroxypropane-1-carboxyl-CoA |
| (2R,3S,4S)-2-hydroxypropane-1-carboxyl-CoA |

Figure 43: Screenshot of result for substructure search
5 Conclusion and Future Work

This work describes an unique structure search tool that was built for the benefit of users across all realms. The structure search tool delivers a complete package of the important structure search methods. The most important part of this tool is the isomer search method which allows the user to search for major sub-categories of isomers. This property gives this tool an edge over the other structure search tools available as most of them only provide options for stereoisomer and tautomer search. The substructure search method allows the user to search for structures with one to five substructures. The isoconnected search method will retrieve structures with exact stereochemistry as the query structure. It can also retrieve structures which are isoconnected with the query structure but without matching stereochemistry. The general structure search will provide the user with information about different chemical properties of structures.

The graph grammar in the Klotho database has proved to be an efficient way of storing information for chemical compounds. The high level language or config rule and the low level language or term-form both gave complimentary information about every compound. While the config rule gave us information about the overall architecture of the compound and the main fragments involved, the term form gave detailed information about every atom in the molecule taking into consideration each atom and all the attributes related to that atom like bond nature, bond direction, chirality etc. This stored stereochemical information enabled the efficient development of the structure search methods described in this work.

The search methods in this work used efficient algorithms to ensure maximum reduction of search space for faster computation time. The use of heuristics with different types of descriptors considerably reduced the search space and got rid of the irrelevant compounds for the particular query. And the precomputation of these descriptors saved computation time and cost each time a query was executed. After the initial screening process by the use of heuristics, atom by atom mapping match was performed followed by a backtracking algorithm to ensure an exhaustive search. For the isomer search method, the algorithms constructed from definitions of each isomer type were able to retrieve accurate results. The flow diagram Figure 44, was constructed to show the overall heuristics of the the search methods isomer search, substructure search and isoconnected structure
There is a lot of room for future work in this project. A whole informational database can be built with the search results from Klotho, giving information about what compounds have what kind of substructures. This would help in relating chemical activity within compounds and be beneficial to users from different fields of work. Another interesting add-on would be to be able to search for conformational isomers. This would make the current isomer search method essentially complete.
Figure 44: Overview of all the structure search methods combined together.


APPENDIX
Table 12: Symbol table for algorithms

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>$q$</td>
<td>Query structure</td>
</tr>
<tr>
<td>$D$</td>
<td>Set of descriptors</td>
</tr>
<tr>
<td>$c$</td>
<td>Compound</td>
</tr>
<tr>
<td>$C$</td>
<td>Set of compounds</td>
</tr>
<tr>
<td>$F$</td>
<td>Formula</td>
</tr>
<tr>
<td>$Ct$</td>
<td>Compound type</td>
</tr>
<tr>
<td>$Cl$</td>
<td>Class</td>
</tr>
<tr>
<td>$Kt$</td>
<td>Keto-enol type</td>
</tr>
<tr>
<td>$Db$</td>
<td>Multiple bonds</td>
</tr>
<tr>
<td>$H$</td>
<td>H atoms</td>
</tr>
<tr>
<td>$Tk$</td>
<td>Keto-enol tautomer</td>
</tr>
<tr>
<td>$Rc$</td>
<td>Ring-chain type</td>
</tr>
<tr>
<td>$Tr$</td>
<td>Ring-chain tautomer</td>
</tr>
<tr>
<td>$Fn$</td>
<td>Functional group</td>
</tr>
</tbody>
</table>

for input query structure $q$ and a set of descriptors $D$

$\forall c \in C$ retrieve set $C_{D_i}$ with descriptors $D_i$ where $D_i \in D$

if $C_{D_i} \neq \emptyset$ then
    compute formula $F$ for $q$
    retrieve $C_{D_i,F}$ for all $c \in C_{D_i}$ with formula $F_i$ where $F_i = F$
    if $C_{D_i,F} \neq \emptyset$ then
        for all $c \in C_{D_i,F}$ retrieve $C_{D_i,FCl_i}$ with compound type $Ct_i$ same as compound type of $q = Ct_q$  
        if $C_{D_i,FCl_i} \neq \emptyset$ then
            for all $c \in C_{D_i,FCl_i}$ retrieve $C_{D_i,FCl_i,Ct_i}$ with class $Cl_i$ different as class of $q = Cl_q$
            if $C_{D_i,FCl_i,Ct_i} \neq \emptyset$ then
                all $c \in C_{D_i,FCl_i,Ct_i}$ is a functional isomer of $q$
            else if $C_{D_i,FCl_i,Ct_i} = \emptyset$ then
                no functional isomer for $q$
            end if
        else if $C_{D_i,FCl_i} = \emptyset$ then
            no functional isomer for $q$
        end if
    else if $C_{D_i,F} = \emptyset$ then
        no functional isomer for $q$
    end if
else if $C_{i,D} = \emptyset$ then
    no functional isomer for $q$
end if

Figure 45: Algorithm 6 for functional isomer search
for input query structure \( q \) and a set of descriptors \( D \)
\( \forall c \in C \) retrieve set \( C_{D_i} \) with descriptors \( D_i \) where \( D_i \in D \)

if \( C_{D_i} \neq \emptyset \) then
  compute formula \( F \) for \( q \)
  retrieve set \( C_{D,F} \) for all \( c \in C_{D_i} \) with formula \( F_i \) where \( F_i = F \)
  if \( C_{D,F} \neq \emptyset \) then
    retrieve set \( C_{D,F,Cl} \) for all \( c \in C_{D,F} \) where class \( Cl_i \) is different than class of \( q = Cl_q \)
    if \( C_{D,F,Cl} \neq \emptyset \) then
      remove multiple bonds and retrieve set \( C_{D,F,Cl,H} \) for all \( c \in C_{D,F,Cl} \)
      remove H atoms and retrieve set \( C_{D,F,Cl,H,KT} \) for all \( c \in C_{D,F,Cl,H} \)
    end if
  end if
else if \( C_{D,F} = \emptyset \) then
  no tautomer for \( q \)
end if

else if \( C_{i,D} = \emptyset \) then
  no tautomer for \( q \)
end if

\( \emptyset = i,D \neq i \) to gain in multiple bond in \( q \)

\( i \neq R \) where class of \( q \) is different than class of \( q = Cl_q \)

\( i \neq Kt \) where structure is a substructure of \( q_{Kt} \)

Figure 46: Algorithm 7 for tautomer search
for input query structure $q$ and a set of descriptors $D$

$\forall c \in C$ retrieve set $C_D$ with descriptors $D_i$ where $D_i \in D$

if $C_D \neq \emptyset$ then

compute formula $F$ for $q$

retrieve $C_D, F$ for all $c \in C_D$ with formula $F_i$ where $F_i = F$

if $C_D, F \neq \emptyset$ then

for all $c \in C_D, F$ retrieve $C_D, F, Ct$ with compound type $Ct_i$ same as compound type of $q = Ct_q$

if $C_D, F, Ct \neq \emptyset$ then

for all $c \in C_D, F, Ct$ retrieve $C_D, F, CtCl$ with class $Cl_i$ same as class of $q = Cl_q$

if $C_D, F, CtCl \neq \emptyset$ then

remove functional group $Fn$ in $q$ to retrieve $q_{new}$

for all $c \in C_D, F, CtCl$ remove $Fn$ and retrieve set $C_D, F, CtClFn'$

if $C_D, F, CtClFn' \neq \emptyset$ then

for all $c \in C_D, F, CtClFn'$ retrieve set $C_D, F, CtClFn'B$ where current structure of $c$ is not same as structure of $q_{new}$

if $C_D, F, CtClFn'B \neq \emptyset$ then

all $c \in C_D, F, CtClFn'B$ is a skeletal isomer of $q$

else if $C_D, F, CtClFn'B = \emptyset$ then

no skeletal isomer for $q$

end if

else if $C_D, F, CtClFn' = \emptyset$ then

no skeletal isomer for $q$

end if

else if $C_D, F, CtCl = \emptyset$ then

no skeletal isomer for $q$

end if

else if $C_D, F = \emptyset$ then

no skeletal isomer for $q$

end if

else if $C_i, F = \emptyset$ then

no skeletal isomer for $q$

end if

Figure 47: Algorithm 8 for skeletal isomer
for input query structure $q$ and a set of descriptors $D$
\forall c \in C$ retrieve set $C_{D_i}$ with descriptors $D_i$ where $D_i \in D$

if $C_{D_i} \neq \emptyset$ then
  compute formula $F$ for $q$
  retrieve $C_{D_i,F}$ for all $c \in C_{D_i}$ with formula $F_i$ where $F_i = F$

if $C_{D_i,F} \neq \emptyset$ then
  for all $c \in C_{D_i,F}$ retrieve $C_{D_i,F,Ct_i}$ with compound type $Ct_i$ same as compound type of $q = Ct_q$
  if $C_{D_i,F,Ct_i} \neq \emptyset$ then
    for all $c \in C_{D_i,F,Ct_i}$ retrieve $C_{D_i,F,Ct_i,Cl_i}$ with class $Cl_i$ same as class of $q = Cl_q$
    if $C_{D_i,F,Ct_i,Cl_i} \neq \emptyset$ then
      find position $Pos_{F_n}$ of functional group $F_n$ in $q$
      for all $c \in C_{D_i,F,Ct_i,Cl_i}$ retrieve set $C_{D_i,F,Ct_i,Cl_i,Pos_{F_n},i}$ where position of functional group $F_n \neq Pos_{F_n}$
      if $C_{D_i,F,Ct_i,Cl_i,Pos_{F_n},i} \neq \emptyset$ then
        all $c \in C_{D_i,F,Ct_i,Cl_i,Pos_{F_n},i}$ is a positional isomer of $q$
      else if $C_{D_i,F,Ct_i,Cl_i,Pos_{F_n},i} = \emptyset$ then
        no positional isomer for $q$
    end if
  else if $C_{D_i,F,Ct_i} = \emptyset$ then
    no positional isomer for $q$
  end if
else if $C_{D_i} = \emptyset$ then
  no positional isomer for $q$
end if
else if $C_{D_i,F} = \emptyset$ then
  no positional isomer for $q$
end if
else if $C_{i,D} = \emptyset$ then
  no positional isomer for $q$
end if

Figure 48: Algorithm 9 for positional isomer
for input query structure $q$ and a set of descriptors $D$
\forall c \in C$ retrieve set $C_{D_i}$ with descriptors $D_i$ where $D_i \in D$

if $C_{D_i} \neq \emptyset$ then

compute formula $F$ for $q$
retrieve $C_{D_i,F}$ for all $c \in C_{D_i}$ with formula $F_i$ where $F_i = F$

if $C_{D_i,F} \neq \emptyset$ then

for all $c \in C_{D_i,F}$ retrieve $C_{D_i, FE}$ with structure exactly same as $q$

if $C_{D_i, FE} \neq \emptyset$ then

check for direction $d$ in structure rule of $q$

if $d = cis$ then

for all $c \in C_{D_i, FE}$ retrieve $C_{D_i, FETr}$ with direction $d = trans$

if $C_{D_i, FETr} \neq \emptyset$ then

all $c \in C_{D_i, FETr}$ is a geometrical isomer of $q$

else if $C_{D_i, FETr} = \emptyset$ then

no geometrical isomer of $q$

end if

end if

end if

if $d = trans$ then

for all $c \in C_{D_i, FE}$ retrieve $C_{D_i, FECs}$ with direction $d = cis$

if $C_{D_i, FECs} \neq \emptyset$ then

all $c \in C_{D_i, FECs}$ is a geometrical isomer of $q$

else if $C_{D_i, FECs} = \emptyset$ then

no geometrical isomer of $q$

end if

end if

else if $C_{D_i, FE} = \emptyset$ then

no exact structure match for $q$

end if

else if $C_{D_i, F} = \emptyset$ then

no exact structure match for $q$

end if

else if $C_{i,D} = \emptyset$ then

no exact structure match for $q$

end if

Figure 49: Algorithm 10 for geometric isomer search
for input query structure \( q \) and a set of descriptors \( D \)
\forall c \in C \text{ retrieve set } C_{Di} \text{ with descriptors } D_i \text{ where } D_i \in D

\textbf{if } C_{Di} \neq \emptyset \textbf{ then}

compute formula \( F \) for \( q \)
\text{ retrieve } C_{D,F} \text{ for all } c \in C_{Di} \text{ with formula } F_i \text{ where } F_i = F

\textbf{if } C_{D,F} \neq \emptyset \textbf{ then}

for all \( c \in C_{D,F} \text{ retrieve } C_{Di,F} \text{ after atom by atom matching with structure exactly same as } q \)

\textbf{if } C_{D,F} \neq \emptyset \textbf{ then}

for all \( c \in C_{D,F} \text{ retrieve } C_{D,FCh} \text{ having chiral atoms} \\
\textbf{if } C_{D,FCh} \neq \emptyset \textbf{ then}

for all \( c \in C_{D,FCh} \text{ retrieve } C_{D,FChEv} \text{ having even number of chiral atoms} \\
\textbf{if } C_{D,FChEv} \neq \emptyset \textbf{ then}

for all \( c \in C_{D,FChEv} \text{ separate chiral atoms into two lists} \\
\text{ for all } c \in C_{D,FChEv} \text{ reverse upper half of chiral atoms} \\
\text{ for all } c \in C_{D,FChEv} \text{ compare bond orientation around chiral atoms in both lists} \\
\text{ for all } c \in C_{D,FChEv} \text{ retrieve } C_{D,FChEvMs} \text{ having same bond orientation in both lists} \\
\textbf{if } C_{D,FChEvMs} \neq \emptyset \textbf{ then}

all \( c \in C_{D,FChEvMs} \text{ is a meso-isomer of } q \)
\textbf{else if } C_{D,FChEvMs} = \emptyset \textbf{ then}

no meso-isomer of \( q \)

\textbf{end if}

\textbf{else if } C_{D,FChEv} = \emptyset \textbf{ then}

no meso-isomer of \( q \)

\textbf{end if}

\textbf{else if } C_{D,FCh} = \emptyset \textbf{ then}

no meso-isomer of \( q \)

\textbf{end if}

\textbf{else if } C_{D,FE} = \emptyset \textbf{ then}

no meso-isomer of \( q \)

\textbf{end if}

\textbf{else if } C_{D,F} = \emptyset \textbf{ then}

no meso-isomer of \( q \)

\textbf{end if}

\textbf{else if } C_{i,D} = \emptyset \textbf{ then}

no meso-isomer of \( q \)

\textbf{end if}

Figure 50: Algorithm 11 for meso-isomer search
for input query structure $q$ and a set of descriptors $D$

∀ $c \in C$ retrieve set $C_{Di}$ with descriptors $D_i$ where $D_i \in D$

if $C_{Di} \neq \emptyset$ then
  compute formula $F$ for $q$
  retrieve $C_{Di,F}$ for all $c \in C_{Di}$ with formula $F_i$ where $F_i = F$
  if $C_{Di,F} \neq \emptyset$ then
    for all $c \in C_{Di,F}$ retrieve $C_{Di,F,E}$ after atom by atom matching with structure exactly same as $q$
    if $C_{Di,F,E} \neq \emptyset$ then
      for all $c \in C_{Di,F,E}$ retrieve $C_{Di,F,E,ch}$ having chiral atoms
      if $C_{Di,F,E,ch} \neq \emptyset$ then
        for all $c \in C_{Di,F,E,ch}$ retrieve set $O_c$ with orientation of substituent groups around chiral atoms
        retrieve set $O_{c'}$ with orientation of substituent groups around same chiral atoms in $q$
        for all $c \in C_{Di,F,E,ch}$ retrieve $C_{Di,F,E,ch,En}$ having orientations in $O_c$ exactly opposite to orientations in $O_{c'}$
        if $C_{Di,F,E,ch,En} \neq \emptyset$ then
          all $c \in C_{Di,F,E,ch,En}$ is an enantiomer of $q$
        else
          no enantiomer of $q$
        end if
      else if $C_{Di,F,E,ch} = \emptyset$ then
        no enantiomer of $q$
      end if
    else if $C_{Di,F} = \emptyset$ then
      no enantiomer of $q$
    end if
  else if $C_{Di,F} = \emptyset$ then
    no enantiomer of $q$
  end if
else if $C_{i,D} = \emptyset$ then
  no enantiomer $q$
end if

Figure 51: Algorithm 12 for enantiomer search
for input query structure \( q \) and a set of descriptors \( D \)
∀\( c \in C \) retrieve set \( C_{Di} \) with descriptors \( D_i \) where \( D_i \in D \)
if \( C_{Di} \neq \emptyset \) then
compute formula \( F \) for \( q \)
retrieve \( C_{Di,F} \) for all \( c \in C_{Di} \) with formula \( F_i \) where \( F_i = F \)
if \( C_{Di,F} \neq \emptyset \) then
for all \( c \in C_{Di,F} \) retrieve \( C_{Di,FE} \) after atom by atom matching with structure exactly same as \( q \)
if \( C_{Di,FE} \neq \emptyset \) then
for all \( c \in C_{Di,FE} \) retrieve \( C_{Di,FECh} \) having chiral atoms
if \( C_{Di,FECh} \neq \emptyset \) then
for all \( c \in C_{Di,FECh} \) retrieve set \( O_q \) with orientation of substituent groups around chiral atoms
retrieve set \( O_c \) with orientation of substituent groups around same chiral atoms in \( q \)
for all \( c \in C_{Di,FECh} \) retrieve \( C_{Di,FEChDia} \) having orientations in \( O_c \) not exactly opposite to orientations in \( O_q \)
if \( C_{Di,FEChDia} \neq \emptyset \) then
for all \( c \in C_{Di,FEChDia} \) retrieve \( C_{Di,FEChDia1} \) where an oxygen atom is part of a ring and is attached to \( Ch1 \)
if \( C_{Di,FEChDia1} \neq \emptyset \) then
for all retrieve \( C_{Di,FEChDia1An} \) where an oxygen atom is part of a ring and is attached to \( Ch1 \)
if \( C_{Di,FEChDia1An} \neq \emptyset \) then
all \( c \in C_{Di,FEChDia1An} \) is an anomer of \( q \)
else if \( C_{Di,FEChDia1} = \emptyset \) then
no anomers of \( q \)
all \( c \in C_{Di,FEChDia1} \) is an epimer of \( q \)
end if
else if \( C_{Di,FEChDia1} = \emptyset \) then
no epimer or anomer of \( q \)
end if
for all \( c \in C_{Di,FECh} \) retrieve \( C_{Di,FEChDia2} \) having orientations in \( O_c \) different than orientations in \( O_q \) for more than one chiral atom
if \( C_{Di,FEChDia2} \neq \emptyset \) then
all \( c \in C_{Di,FEChDia2} \) are other diastereomers of \( q \)
else if \( C_{Di,FEChDia1} = \emptyset \) then
no other diastereomers of \( q \)
end if
else if \( C_{Di,FEChDia1} = \emptyset \) then
no diastereomer of \( q \)
end if
else if \( C_{Di,FECh} = \emptyset \) then
no diastereomer of \( q \)
end if
else if \( C_{Di,FE} = \emptyset \) then
no diastereomer of \( q \)
end if
else if \( C_{Di,F} = \emptyset \) then
no diastereomer of \( q \)
end if
end if

Figure 52: Algorithm 13 for diastereomer search