Efficient Discovery of Common Substructures in Macromolecules *

Srinivasan Parthasarathy and Matt Coatney
Computer and Information Science
Ohio State University
{srini,coatney}@cis.ohio-state.edu
Tech Report OSU-CISRC-8/02-TR20

1 Abstract

Biological macromolecules play a fundamental role in both life and many of the diseases that plague life; as such, they are of great interest to fields such as drug discovery and chemical genomics. Yet due to macromolecules' large size, complexity, and the inability to obtain good structural data, development of efficient and effective data mining techniques for elucidating exact structure- function macromolecular relationships has been ill explored. Previous macromolecule techniques have focused primarily on sequence analysis, which does not account for true three-dimensional structure or sequence fluctuations due to evolutionary divergence. Furthermore, traditional chemical substructure discovery techniques that utilize true three-dimensional structural information are incapable of scaling to such large sizes or handling noise. In this paper, we present a novel scalable approach to efficiently discovering substructures in macromolecules based on only the three-dimensional structural coordinate data, without the need for in-depth domain-specific knowledge. The approach combines a structure-based frequent pattern identification algorithm with techniques and optimizations to both reduce search space and handle coordinate noise. We analyze the computational performance of this approach and its optimizations compared to more traditional substructure discovery approaches. We then validate that the approach can discover meaningful substructures in noisy macromolecule data by automated discovery of the peptide backbone, α-helices, and β-sheets for several proteins. Finally, we show that the technique is more robust than traditional sequence-based approaches at determining structural, and thus functional, similarity between proteins.

* This work was partially supported by an Ameritech Faculty Fellowship.

2 Introduction

In recent years, data mining techniques have been extended beyond traditional domains such as business and marketing to apply to the physical science arena. Many of the hallmark characteristics of data mining, including detection of important patterns, handling of large search spaces, and coping with noisy datasets, are well-suited for the unique challenges presented by bio- and chemo-informatics domains. The substructure discovery sub-domain of bioinformatics is a classic example of the need for powerful yet efficient tools to discover previously unknown patterns. The motivation for locating substructures is that the structure of a particular molecule determines its biochemical function. This structure-function relationship is critical to the understanding the mechanisms of drug activity, toxicity, and disease.

Previous research [5] [6] [8] has examined frequent substructures in large databases of small chemical compounds; this research is conducive to areas such as toxicology and drug discovery. Recently, macromolecules such as proteins and nucleic acids have received increased attention in data mining [12] [16] [20], as these biological molecules are the core components of life and are the targets of many drugs. Most existing macromolecule research has focused on sequence analysis of polymer substituents such as nucleotide bases and amino acids, but this approach at best only approximates the structure-function relationship.

What is needed is an efficient, scalable, and noise-handling substructure mining approach that can exactly determine three-dimensional structural features for use in structure-function research. Macromolecule substructure mining is similar in approach to techniques for small chemicals; however, distinct differences warrant further consideration. These differences include the large number of atoms in the average macromolecule, the large size of interesting substructures, and a high amount of noise in three-dimensional macromolecule data as a consequence of current X-ray crystallography technology. Macromolecule size presents a truly daunting search space; macromolecules can have thousands of atoms, each of which is analogous to an object in a spatial database. Even smaller combinations of these atoms, known as atomsets (similar to itemsets in transaction-based data mining), can lead to a computationally infeasible search space. For instance, a macromolecule containing 5000 atoms contains 12.5 million 2-atomsets and 20 billion 3-atomsets. This issue is exacerbated by the fact that interesting patterns in macromolecules often contain at least five atoms and perhaps twenty or more. This is due to the fact that macromolecules are polymers, consisting of repeated similar subcomponents. Thus, interesting patterns are typically particular conformations and sequences of these subcomponents.

This paper presents a scalable method that combines traditional frequent pattern techniques with search space and noise-handling optimizations to efficiently identify *internal* substructures in *large* coordinate datasets. One of the key advantages of our approach is its relative independence from detailed chemical domain knowledge. While this algorithm was designed specifically to address issues related to macromolecules, it is capable of efficiently identifying substructures in any spatial coordinate data, without the need for in-depth scientific understanding of the dataset's properties. The search space is dramatically reduced through range pruning, where atom pairs outside a certain range are ignored. Such an optimization is possible because atom-atom interaction is inversely related to distance. A structure-based version of frequent pattern discovery further reduces the atom search space. Combining these two techniques allows efficient construction of larger substructures. Furthermore, we employ an approximate counting mechanism and distance binning to minimize the effects of coordinate data noise, and we present several performance enhancements to the discovery algorithm. Experimental results on protein data show that this approach applied to intra-molecule analysis is capable of successfully and efficiently identifying the peptide backbone, α-helices, and β-sheets of several proteins. This serves as a validation that our technique can indeed find meaningful substructures without protein-specific chemical knowledge; with more research we hope to discover novel substructures in the same manner we rediscovered existing protein substructures. Finally, we show that the approach is more robust than traditional sequence-based approaches at determining structural similarity of proteins, including high structural similarity of Ribonuclease A from two evolutionarily diverse species.

The ideas presented in this paper form the first part of an exact structure- function macromolecular data mining process. Frequent substructures identified by this approach can then be fed into various data mining techniques, including classification [18], clustering [9], and sequence mining [17] algorithms to produce structure-derived macromolecule functional classes for proteins, genes, and auxiliary RNAs. This automated approach to functional class generation not only dramatically reduces the time and effort spent by manually developing such classes, the technique also improves the quality of the resulting classes, since these classes are formed directly from the structures of the proteins and thus adhere to the structure-function principle. In addition, once classes have been constructed, a novel protein's function can be predicted based only on its three-dimensional structure.

We begin in section 3 by presenting related work. We then present the general algorithm in section 4, followed by various optimizations for addressing issues related to studying macromolecules. Finally, we present experimental results for both performance and accuracy in section 6.

3 Related Work

3.1 Frequent Pattern Mining

Many of the techniques recently developed for efficient data mining of frequent patterns are applicable to substructure discovery in molecules. The Apriori algorithm presented in [1] dramatically reduces the search space of a database through use of anti-monotone frequency constraints. The algorithm combines smaller patterns to produce only those candidate larger patterns that can possibly be frequent. Of particular interest to the macromolecule domain is the related AprioriTID algorithm [1]. In this algorithm, patterns and their origin are stored in memory and used to produce and prune future patterns, rather than requiring a re-scan of the database; this can be more efficient for large, complex datasets whose number of patterns greatly reduces with increased pattern size.

Frequent pattern techniques that analyze spatial data and sequences are also relevant, but they are distinctly different from this work in that many rely heavily on domain knowledge and do not adequately address scalability and noise effect minimization. Recent research has examined spatial data for such domains as cartography, network topography, and geographical information systems [13]. Sequence analysis and episodes have been analyzed as well by [2] [3] [15].

3.2 Substructure Discovery

Application of data mining to discover frequent substructure patterns in three-dimensional graphs is not novel; however, previous research has primarily targeted small molecules and has not addressed issues of scalability and minimization of noise effects. The work of [8] detects substructures of three-dimensional graphs, but the algorithm does not consider atom type, which due to steric and electrostatic behavior is critical to the quality of discovered substructures. [5] more explicitly targets the chemical domain by considering atom and bond types as well as background biological information. However, this approach appears to have scalability issues for even small toxicological compounds let alone for macromolecules. [6] presents a more general approach to substructure discovery, and of particular interest is its study of graph compression and interestingness of discovered substructures with and without domain knowledge bias. Similarly, [19] presents a general substructure-based framework and combines several data mining techniques to improve the robustness of substructure discovery.

As mentioned previously, most macromolecular analysis has focused primarily on sequence analysis. For example, [10] analyzes a small group of PDB protein data for substructure patterns through sequences of proximal amino acids. This technique does consider some meta-structure information based on distances between amino acids, but it still does not fully model structural aspects of the proteins. [11] presents a powerful method for obtaining useful meta-structural information when obtaining coordinate data from crystallography is infeasible. This approach complements the work presented here.

This paper differs from previous substructure discovery techniques in that it analyzes the occurrence of true three-dimensional frequent substructures *within a single large molecule* rather than across a series of small molecules, and it provides exact structural features that tend to have more function-predictive power than sequence-based techniques. We extend the ideas presented in [14], providing a more formal analysis of the problem and presentation of the algorithm, several performance optimizations, and quantified experimental results for the algorithm and optimizations. This paper also extends the analysis of secondary structure identification to include a β-sheet case study and includes research on using substructure fingerprints for finding similarity between proteins.

4 Algorithm

4.1 Substructure Representation

Molecules and their substructures are often described as three-dimensional coordinate graphs, where atoms are nodes and chemical bonds are edges. Examples include the MDL MOL format and the Protein Database PDB format. Substructures are subgraphs of overall coordinate graphs, normalized to account for varying orientations in space. Two substructures are considered equal if, after an arbitrary number of spatial translations on one substructure, both substructures are described by the same graph.

The type of connection between two atoms, known as the chemical bond type (single, double, or triple), can be ignored without loss of information, since the bond type can be inferred from the types of the two atoms and the distance between them. Furthermore, we consider all proximal atom relationships rather than just connections. This allows for better detection of non-covalent interractions and complex active sites, both of which involve spatial rather than connection relationships.

We define a new concept, the *atomset*, that captures all information of a three-dimensional graph in a form that facilitates quick comparison without the need for coordinate translation. We store three-dimensional information between a pair of atoms, A_i and A_j, in a *mining bond*. The mining bond $M(A_iA_j)$ is a 3-tuple of the form

$$M(A_iA_j) = \{A_i type, A_j type, distance(A_iA_j)\}$$

A k-atomset X, which is a substructure containing k connected atoms, is then defined as a tuple of the form

$$X = \{\mathbf{S_X}, A_1, A_2, \ldots, A_k\},$$

where A_i is the i^{th} atom and $\mathbf{S_X}$ is the set of mining bonds describing the atomset. By defining atom pair combinations with mining bonds, the three-dimensional graph is completely represented in a redundant form, such that two atomsets X and Y are considered to be the same chemical substructure if $\mathbf{S_X} = \mathbf{S_Y}$. While stereochemistry is not explicitly handled by this representation, it can be implicitly handled by appending a chirality label (e.g. L or R) to the atom type, such that different stereoisomers produce different, non-equivalent atomsets.

In a naive approach, $\mid \mathbf{S_X} \mid$ is equal to $\binom{k}{2}$, which represents all possible atomset permutations. We next introduce the concept of range pruning, which allows us to not only dramatically reduce the search space of the algorithm but also significantly reduce the number of mining bonds needed to describe a given atomset.

4.1.1 Range Pruning

An exhaustive analysis of possible atom combinations, even at lower levels, is computationally infeasible for large graphs such as macromolecules. Application of chemical domain knowledge through range pruning affords a great reduction in search space and allows us to fully describe the three-dimensional representation of a substructure with fewer mining bonds. Range is a user-specified constraint defined as the maximum allowable Euclidean distance between two atoms for them to be considered an atom pair. Range pruning reduces the search space to only consider those atom-atom interactions that are chemically meaningful.

This optimization relies on the fact that the associated energy between any two atoms in a molecule is inversely related to the inter-atom distance. Beyond a certain range, atom-atom interaction is negligible and the two atoms can be considered independent of one another. Thus, we can limit the space of atom pairs for candidate 2-atomset generation to within a certain range, ignoring those atom combinations outside of this range.

The range of interaction differs depending on the atom types involved. For bio-molecules, the predominant atom by far is carbon, so we ensure that the range sufficiently describes possible carbon-carbon interactions. The typical carbon-carbon single bond has a bond length of $1.54\ \overset{\circ}{A}$, with double and triple bonds having shorter length. We have found that a range of $4.5\ \overset{\circ}{A}$, roughly three times a carbon-carbon single bond, is sufficient for encompassing all possible carbon-carbon interactions [14]. This range not only accounts for direct bonds but also allows for indirect relationships, such as atom-atom spatial proximity and hydrogen-bonding. Most macromolecular coordinate datasets do not include hydrogen atoms due to limitations of structure determination techniques, yet hydrogen-bonding plays a significant role in the three-dimensional structure and functionality of large bio-molecules. A suitable range can handle such limitations in the raw data.

While we fully enforce the range restriction on initial atom pairs (creation of 2-atomsets), we relax this restriction when building up larger substructures such that two atoms may be included in an atomset even if they do not satisfy the range constraint, so long as they share common atoms that do meet the range constraint. For instance, if we have two atomsets X and Y with atoms (A_1, A_2, A_3) and (A_2, A_3, A_4) respectively, we allow the creation of a candidate atomset Z with atoms (A_1, A_2, A_3, A_4) even if distance$(A_1, A_4) \geq$ range.

Without such a relaxation, higher range values would be needed to detect large substructures; this in turn would limit range pruning's ability to reduce the search space, which is a key optimization of this algorithm. Also, such a relaxation allows us to reduce the number of mining bonds needed to fully describe the atomset, since atoms not directly joined by mining bonds are still indirectly associated through mining bonds with their shared atoms. This reduction in necessary mining bonds significantly improves both computational performance and memory efficiency.

4.2 Generating Incremental Atomsets

Our approach to generating incremental k-atomsets from two (k-1)-atomsets is similar in concept to frequent pattern discovery algorithms. However, our approach is distinctly different from traditional techniques in that removal of internal elements can cause the resulting substructures to become infrequent, based on the range constraint. We describe here a range-based anti-monotone frequency restriction for the chemical domain.

Definition 1 *Extremal atoms for a given k-atom structure are the two or more atoms that are furthest away from each other within the structure.*

Definition 2 *Extremal (k-1)-atom substructures for a given k-atom substructure are those substructures that contain at least one of the extremal atoms.*

By definitions 1 and 2, each structure has at least two extremal substructures. These definitions lead to the statement of Lemma 1, which will help prune the number of potentially frequent patterns (candidate atomsets) that will need to be evaluated.

Lemma 1 *Any frequently occurring k-atom structure has* at least *two (k-1)-atom substructures that are frequent and that satisfy the input range criteria (R)*

Corollary 1 *For any frequently occurring k-atom structure all of its extremal substructures are frequent and satisfy the input range criteria (R)*

4.2.1 Proof Sketch

It is trivial to show that all substructures of a given frequent substructure must be frequent. However not all of these substructures will satisfy the range criteria. A simple example that shows why this is so involves a line of points each separated by a distance corresponding to the range. If one takes out any of the points except the two end points, the resulting substructure will not satisfy the range criteria, because eliminating such a point breaks the linkage. The rest of the proof is based on the fact that eliminating either of the extremal points cannot possibly break the linkage.

Now assume we are given the set S of frequently occurring (k-1)-atomsets. By the above lemma and its corollary, the set C of potentially frequent k-atomsets is limited to those candidates whose extremal substructures are in S. In essence, if the extremal substructures are not in S then the potential candidate can never be frequent.

4.3 Candidate Generation and Pruning

Candidate generation and pruning of (k+1)-atomsets from frequent k-atomsets requires special consideration. In a standard frequent pattern discovery approach, (k+1)-sets are constructed from k-sets without regard to the source from which the sets originated. Infrequent (k+1)-sets are then pruned by re-scanning the source and counting the number of 0 for each (k+1)-set. This approach in a structural context is computationally very expensive, due to the vast number of possible structural permutations in a typical molecule and particularly macromolecules. Rather, candidate generation and pruning are performed using only the available atomsets and do not re-scan the entire molecule.

(k+1)-atomsets are formed using only frequent k-atomsets; this approach is possible since atomset frequency is anti-monotone. Rather than simply generating incremented substructure *patterns*, which would then be used to query the molecule for frequency, *all* instances of (k+1)-atomsets are generated by combining the atoms of frequent

1. define set $\mathbf{C}$; Incremented candidate atomsets
2. **for** all $atomset_i$ in frequent k-atomsets
3. **for** all $atomset_j$ in frequent k-atomsets: $j > i$
4. Set $\mathbf{D} = (atom_i \notin atoms_j \forall atoms_i) \cup (atom_j \notin atoms_i \forall atoms_j)$; Find atoms different
5. if $| \mathbf{D} |= 2$ and if $\forall atom_x \in atomset_i \neq \mathbf{D_i}$, distance$(atom_x, \mathbf{D_j}) \leq$ distance $(\mathbf{D_i}, \mathbf{D_j})$; if two extremal atoms are different
6. $candidateAtomset_{ij} =$ new atomset $(\mathbf{S_i} \cup \{M(A_i \mathbf{D_j})$ for $A_i \in atomset_i$ $| \; distance(A_i, D_j) < range\}$ $, A_1, \ldots, A_k, \mathbf{D_j})$
7. $\mathbf{C} = \mathbf{C} \cup \{candidateAtomset_{ij}\}$; add to set of incremented atomsets

Figure 1: Candidate Atomset Generation Algorithm

1. Prune infrequent atoms (1-atomsets)
2. Generate candidate 2-atomsets from frequent atoms
3. Prune infrequent 2-atomsets
4. $k = 3$
5. **while** ($|$ frequent k-atomsets $| > 0$)
6. Generate candidate k-atomsets from frequent (k-1)-atomsets
7. Prune infrequent k-atomsets
8. $k = k + 1$

Figure 2: Common Substructure Discovery Algorithm

k-atomsets. The algorithm for this approach is shown in Figure 1. Since the substructure-atomset relationship is one-to-many, and since atom-based combinations are more complex than mining bond combinations, this candidate generation method is more computationally expensive. However, this expense is more than offset by the simplification in pruning.

Pruning is then accomplished simply by counting the number of atomsets that define, based on their mining bond sets, the same substructure, keeping those whose count is above a user-specified minimum support. This approach is faster than traditional approaches both because counting occurs without the need to go back to the entire molecule and because counting is done through pattern-pattern instead of pattern-dataset matching, which results in far fewer comparisons. Standard pruning is straightforward and can be accomplished simply by hashing atomsets into bins of substructures based on the set of mining bonds. More advanced approaches to pruning are needed when handling noisy data; this will be discussed in Section 5.

4.4 Algorithm Details

The algorithm used here for mining frequent substructures is shown in Figure 2. This algorithm is general enough to handle substructure analysis for any type of molecule; however, the size and noise issues of macromolecules limit the usefulness of this algorithm as-is. In the next section, we present a series of optimizations that improve algorithm performance, handle noisy coordinate data, and utilize domain knowledge to adequately analyze these macromolecules.

5 Optimizations

While the algorithm presented in the previous section is sufficient for mining smaller chemicals, such as those analyzed in drug discovery and toxicological studies, it cannot sufficiently handle the noise and large search space that are characteristic of macromolecules. In this section we present several algorithm optimizations that significantly improve the performance and quality of macromolecule substructure analysis.

5.1 Noise Handling

5.1.1 Distance Binning and Resolution

Macromolecule coordinate data is notoriously noisy and has low resolution. Proteins are often difficult to isolate and crystallize, and even once they are prepared for X-ray crystallography, they maintain a certain fluidity due to internal protein binding with water molecules. One approach for handling this noise is discretization, a common data mining technique [7]. We discretize the raw Euclidean distance between two atoms by binning; a resolution value is chosen that divides the distance into equiwidth bins, represented efficiently as bits in the mining bond. Binning of the data not only simplifies calculations (thus improving performance), but it also handles minor fluctuations in distances due to the low resolution of current macromolecule coordinate technology. Initial studies on a PDB protein dataset suggest that algorithm resolutions between $0.04\ \mathring{A}$ and $0.10\ \mathring{A}$ best minimize noise effects while maintaining meaningful atom-atom relationships. Studies have also found that there is a direct relationship between crystallography resolutions and algorithm resolutions suitable for detecting meaningful substructures.

5.1.2 Recursive Fuzzy Hashing

Due to the high level of noise inherent in current macromolecule structure deduction techniques, strict matching of patterns, even with binning, results in poor performance for substructure discovery. What is needed is a pruning mechanism that relaxes the strict matching criteria such that two atomsets X and Y are considered to be the same chemical substructure if $\mathbf{S_X} \approx \mathbf{S_Y}$. One such approach is recursive fuzzy hashing (RFH), in which atomsets are analyzed one mining bond at a time and atomsets are hashed (using the current mining bond's integer value) both to the exact and neighboring locations to minimize the impact of coordinate noise.

As one might expect, this is significantly more computationally expensive than the standard approach; however, such an approach is essential for effectively minimizing noise effects to allow for identification of larger macromolecule substructures. This will be quantified in the next section; for now, we simply present the technique in

```
1. m = 1    ; Examine first mining bond
2. C = candidateAtomsets    ;
/*Start by examining all candidate atomsets*/
3. for all atomset_i in C
4.    hash(M_m, M_m + 1, M_m − 1 in atomset_i)
5. for all H_i in hash_bins
6.    remove H_i if | H_i | < minSupport
7. m = m + 1
8. while m <=| M |
9.    for all H_i as C
10.      goto step 3
11. frequentAtomsets = H_1 ∪ H_2 ... H_n
```

Figure 3: Recursive Fuzzy Hash Pruning Algorithm

Figure 3.

RFH produces a top-down hash tree (see [4] for a thorough description of hash trees), with the root node representing the set of candidate atomsets for the particular k. We recursively split the nodes into several child hash bins, based on the atomsets' mining bond for a particular tree level. For RFH, we hash both to the exact hash location and in its immediate neighborhood ($+/-$ 1 resolution unit). Hash bins whose atomset count is less than a user-specified minimum support are pruned from further consideration. Once the tree is constructed, a new set of pruned atomsets is generated from the remaining leaf nodes.

5.2 Performance

5.2.1 Depth First Pruning

Hash trees produced by RFH are only as deep as the number of mining bonds used to represent the atomsets. In practice, this is often between 10 and 20 levels. The tree is significantly wider, however, due both to the large numbers (thousands or tens of thousands) of atomsets and the use of fuzzy-hashing, which can triple the width of the tree. In order to allow a memory efficient analysis of the tree, particularly at higher substructure sizes, we employ a standard depth-first analysis. This permits us to only look at a small slice of the tree at any given time.

Breadth-first tree analysis must generate all hash bins for a particular level before being able to prune away infrequent bins and free up memory. For very wide trees, this can be quite memory intensive and does not scale well to deeper levels, where the tree width explodes. Depth-first analysis, on the other hand, only analyzes one branch of the tree at a time, which is a fraction of the tree's width.

As an example, let us consider a hash tree generated for hemoglobin. The numbers of pre- and post-pruned bins for each level are shown in Table 1. It is evident that even at lower substructure sizes the tree's width/depth ratio is quite large. For a breadth-first analysis of level 7, the memory footprint includes *all* 15K bins, each including a group

Mining Bond Level	Bins (Pre-pruning)	Bins (Post-pruning)
1	171	104
2	985	444
3	5571	1679
4	18032	3676
5	43778	9089
6	84229	12124
7	91090	14591
8	123403	13324
9	71475	20555

Table 1: Pre- and Post-Pruning Bin Width of RFH Tree

of atomset references. A depth-first memory footprint will have far fewer bins, and thus atomset references, under active consideration. If we assume that there are four general classes of substructures for each hash, then there will be only 12 bins per level (3 per class), giving us approximately 80 bins actively being considered at level 7 at any one time. Thus, for this example the depth-first approach requires $\frac{1}{2}\%$ of the amount of memory the breadth-first approach uses. In larger substructure analyses, this ratio will be even more pronounced. Clearly the depth-first approach is more memory efficient.

While using depth-first analysis on shallow trees is a standard technique, it is worth mentioning because of its importance to macromolecule substructure discovery. The ability to efficiently handle large substructures is critical to the success of macromolecule analysis. It was our experience that without further optimizations the bread-first approach leads to an explosion of the memory space at moderate ($k = 6$) levels, while depth-first analysis maintains a small memory footprint even at higher ($k \geq 9$) levels.

5.2.2 Dynamic Duplicate Screening

The RFH mechanism is not without its drawbacks. Its primary issue is redundant recursive calculations. For example, if a group of atomsets hash to the same three substructure bins and no other atomsets hash into these bins, all three hash bins will produce the same end result. A naive approach to RFH prunes all duplicate substructure bins once the hash tree has been fully formed. However, this leads to a tremendous amount of redundant work.

We instead use a novel approach, dynamic duplicate screening (DDS), that handles duplicates *during* the run. At each level of the hashing algorithm, the set of substructure bins are analyzed, and duplicates are discarded from further consideration. While the analysis itself is expensive, significant time is saved in avoiding redundant calculations. Using this approach, we can gain the noise-handling capability of RFH while lessening the negative impact of duplicate calculations. In addition to speed improvements, this technique also has the benefit of significantly reducing the memory footprint needed for the hash tree, since removal of bins significantly decreases the width of the tree.

Level (k)	$k + k$ Candidate Generation Candidate Atomsets	Exec Time (s)	$k + 2$ Candidate Generation Candidate Atomsets	Exec Time (s)
2	8545	2	8576	2
3	68280	14	73797	24
4	14916	25	113433	70
5	312	28	25630	105

Table 2: Comparison of Candidate Generation Techniques

5.2.3 Analyzing Polymer Backbones

Often, we are primarily interested in the global conformation and super-structures of macromolecules. When this is the case, we can further reduce the search space of a given macromolecule by only considering the polymer's backbone. For proteins, this is the core sequence of amino acid peptides, without the side chains that differentiate amino acids. In essence, we only consider atoms that follow the core thread of a protein; those peripheral atoms that do not directly contribute to this thread are discarded. Such an approach has the advantage of reducing the number of atoms and candidate atomsets through a pre-processing step. We applied this approach to the protein domain to analyze backbone conformations, and the results were promising. As an example, the backbone-only approach identified the same peptide substructures of lysozyme as the full-blown search and ran *five times faster*.

Protein backbones can be further reduced by considering only a single atom per peptide, the α-carbon. While significant micro-substructure information is lost with such an approach, this greatly reduces the protein's search space and allows for more efficient macro-structure detection and description of the overall molecular three-dimensional space. While many traditional approaches for protein description use sequence homology of amino acids, this approach could be used to efficiently classify proteins into functional families based on global three-dimensional shape.

6 Experimental Results

6.1 Performance

Several of the optimizations mentioned in the previous section reduce the effective search space of the algorithm, which in turn significantly improves the algorithm's run time and memory footprint. In this section, we quantify these improvements over the standard, non-optimized approaches.

All experiments were conducted under the Java 1.4 runtime environment on a 4 CPU Sun 420R workgroup server and were allocated 1 GB of memory. The program utilizes multi-threading to take advantage of an SMP architecture.

Range ($\mathring{A}$)	2-atomsets	3-atomsets	Exec Time (s)
4.5	9K	12K	4
6.0	20K	147K	25
9.0	54K	2.3M	536
∞	500K	N/A	N/A

Table 3: Effect of Range Pruning on Search Space

6.1.1 $k + k$ vs. $k + 2$ Candidate Generation

The initial algorithm presented in [14] utilized $k + 2$ candidate generation, in which k-atomsets were combined with 2-atomsets to produce (k+1)-atomsets. This approach was taken prior to full development of the range pruning theory presented in 4.2. With completion of this theory, we are now able to take full advantage of range pruning through $k + k$ candidate generation. $k + k$ combines k-atomsets with themselves for a more restricted superset of the frequent incremented (k+1)-atomsets than could previously be generated by $k + 2$.

Table 2 shows the results of the two generation types against the protein lysozyme. The $k + k$ candidate generation approach is superior to the $k + 2$ approach both in performance and search space reduction. Not surprisingly, the benefit of using $k + k$ generation increases with larger substructure size; this is due to the larger number of possible combinations in the $k + 2$ approach and the increased ability of $k + k$ to restrict atomset permutations. In terms of search space, the $k + k$ approach generates 13% of 4-atomsets produced by $k + 2$ and only 1% of 5-atomsets. The result is a 3-fold improvement in performance at $k = 4$ and a 4-fold improvement at $k = 5$. For the remaining experiments, we utilize only $k + k$ candidate generation.

6.1.2 Range Pruning

Range pruning, as mentioned before, has a dramatic impact on reduction of search space. Table 3 demonstrates this for the protein lysozyme for several different ranges. As expected, search space, and thus run time, decreases as the range used for pruning becomes more restrictive. So long as the range incorporates all real atom-atom interactions of interest, there is no loss of domain-relevant information. Traditional substructure discovery approaches do not consider such range pruning and as such fail to appropriately scale for macromolecules. Clearly, efficient macromolecule analysis is not possible without such a technique.

6.1.3 Dynamic Duplicate Screening

The benefits of the DDS optimization are quite pronounced, particularly at higher levels, due to its ability to handle the exponential growth of redundant calculations. This is illustrated in table 4. The table shows the number of atomsets due to duplicate substructure bins remaining at the end of hash runs for various levels of lysozyme analysis. Clearly, DDS maintains a manageable number of duplicates even at higher levels. On the other hand, the standard approach,

Level (k)	Dynamic Duplicate Screening Duplicates	Exec Time (s)	Standard Duplicates	Exec Time (s)
2	18K	2	19K	2
3	64K	14	81K	15
4	49K	25	312K	26
5	19K	28	2.5M	44

Table 4: Effect of Dynamic Duplicate Screening on Redundant Calculations

	Combined Optimizations			Standard		
Level (k)	Exec Time (s)	Memory (MB)	Candidate Atomsets	Exec Time (s)	Memory (MB)	Candidate Atomsets
2	2	8	9030	2	9	9030
3	24	221	89717	38	257	92524
4	98	152	68770	370	454	335677
5	121	153	10281	1100	1000	183141
6	165	295	1485	Out Of Memory		
7	183	297	147	Out Of Memory		

Table 5: Comparison of Combined Optimizations vs. Standard Algorithm

which screens duplicates only at the end of the run, suffers from an explosive exponential growth rate of duplicate calculations, as evidenced by over 2.5 million atomsets in duplicate substructures during level 5 of the run. For this same level, DDS produces only 19 thousand atomsets, a mere 1% of the standard approach.

The overall performance of a standard approach does not greatly suffer at lower levels, since much of that time is spent in incremental candidate generation. However, as the depth of the hash tree grows and the width grows ever larger, the impressive gains of DDS become evident. For instance, a hemoglobin run to identify α-helices using 9-atomsets took 90 minutes without DDS. *With the optimization, the run took only 5 minutes, resulting in an eighteen-fold increase*. This optimization, when combined with the previously mentioned search space reductions, allows for efficient identification of even large substructures.

6.1.4 Performance of Combined Optimizations

Here we combine all optimizations and compare the performance with traditional structure discovery techniques. The optimized run uses $k + k$ candidate generation and depth-first RFH pruning with DDS. The standard run uses $k + 2$ candidate generation with standard breadth-first RFH. We keep the multi-threading constant, restricting our analysis purely to algorithmic optimizations. The run analyzes the first subunit of hemoglobin (PDB ID: 1BZ0) with minimum support of 70, resolution of 0.06, and range of 4.5. Note that the runs locate the same substructures; they merely differ in their approach to generating candidates and locating these substructures.

Table 5 shows the results for optimized and standard runs. It is immediately evident that the combined optimization approach is capable of efficiently detecting large substructures while maintaining a small memory footprint. On the other hand, the standard run exhibits an explosive search space; this leads to both explosive run time and memory. By

Pruning	Resolution ($\mathring{A}$)	5-Atomsets	Time (s)
Standard	0.06	0	5
Standard	0.18	108	14
Standard	0.1781	125	12
RFH	0.06	128	31

Table 6: Identification of Peptide Substructures in Lysozyme

the end of the fifth level, the standard approach consumes all 1 GB of memory allotted to it; at this point, it would need to begin utilizing the much slower virtual memory, resulting in even poorer performance.

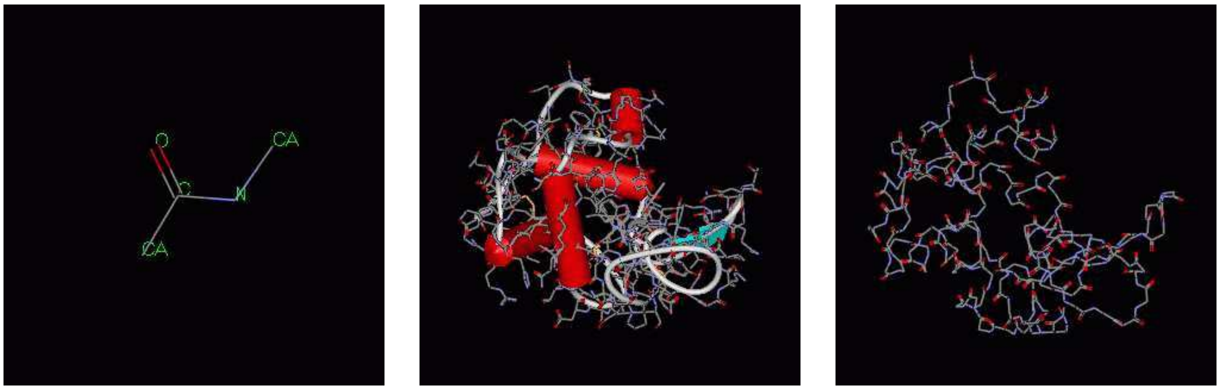
6.2 Substructure Discovery in Proteins

6.2.1 Discovery of the Peptide Backbone

Using the algorithm presented in previous sections, we turn to identifying relevant substructures in protein macromolecules as validation of our technique. We begin by analyzing the 128-amino-acid protein lysozyme (PDB ID: 193L) for 5-atomset peptide substructures. We configure the algorithm with a minimum support of 100 and range of $4.5\ \mathring{A}$. A resolution of $0.06\ \mathring{A}$ was settled on for RFH after attempting a series of values; as mentioned prior, smaller values do not adequately handle protein coordinate noise, and larger values fail to identify appropriate substructures. For instance, a resolution of $0.04\ \mathring{A}$ found 125 5-atomsets, and a resolution of $0.05\ \mathring{A}$ found all but one peptide, which was a trans peptide bond.

Table 6 shows the results of the algorithm using different pruning strategies. RFH significantly increases computation and thus run time when compared to traditional standard hashing techniques. The impact this has on performance can be pronounced, as shown by the large increase in run time. The strength of RFH for handling noise is evident; RFH is able to find all 128 5-atomsets that describe the same peptide substructure. Thus, RFH fully defines lysozyme's peptide backbone. Even with significant modifications to resolution in an attempt to mimic RFH's noise-handling capabilities, the standard approach cannot find all peptides. Above $k = 6$, the standard approach is overcome with coordinate noise and cannot reliably detect substructures. We therefore use the more compute-intensive RFH to further identify interesting protein substructures.

The 5-atomsets found in lysozyme represent one substructure pattern, that of a peptide bond. The substructure is shown in Figure 4. This includes the backbone oxygen and carbon, the α-carbon of residue i and the backbone nitrogen and α-carbon of residue $i + 1$. Combining the substructures, we reconstruct the peptide backbone of lysozyme and compare it with the original protein structure, shown in Figure 4; α-helices are indicated by red cylinders, and β-sheets are indicated by blue arrows.



(a) Peptide substructure (b) 3-dimensional structure (c) Reconstructed Backbone

Figure 4: Peptide backbone reconstruction of Lysozyme

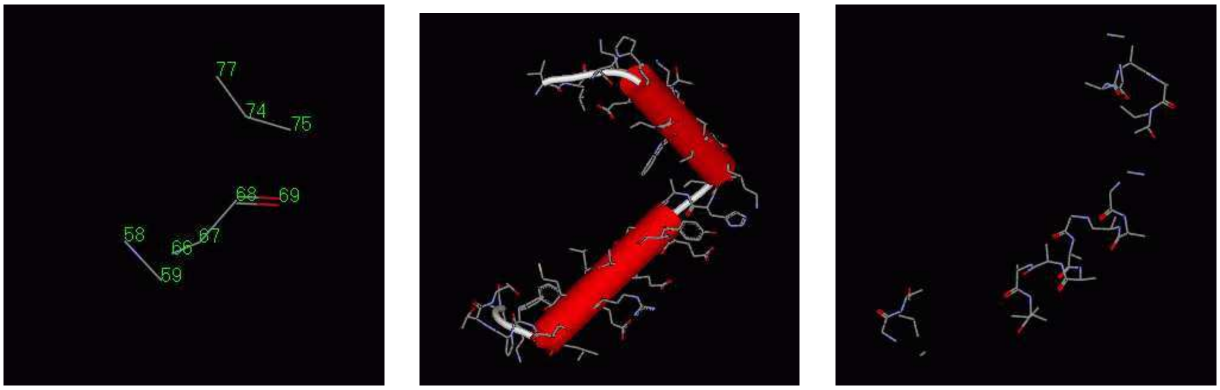
6.2.2 Discovery of α-Helices

We now extend the application of our algorithm to identification of α-helices. For this, we consider the first 141 amino-acid subunit of hemoglobin (PDB ID: 1ZB0). We first attempt to identify smaller substructures that are indicative to alpha helices. We supply the algorithm with a minimum support of 80 and same resolution and range, and we consider 6-atomsets. Two unique substructures consisting of 212 6-atomsets were found (structures omitted due to brevity).

The fist substructure was found to be more frequent than the second. The result seems to indicate that in an α-helix rich protein such as hemoglobin, the β-carbon of residue $i + 1$ is more likely to form a certain conserved structure, as represented by the first substructure. This suggests that protein secondary structures can be detected even at lower substructure sizes. The second substructure appears to simply be an incremented version of the peptide bond and so is of little interest to identification of α-helices.

We next extend the search to 9-atomsets in an attempt to find full α-helix substructures. We apply the algorithm on a smaller section of the hemoglobin subunit, which contains the first two α-helices. We set a minimum support of 15, range of $4.5\ \mathring{A}$, and a resolution of $0.07\ \mathring{A}$.

We discover a single 9-atom substructure, which is composed of atoms from three amino acid residues. The substructure is shown in Figure 5. The substructure contains terminal atoms that are outside the specified range; such a substructure can be formed since we relax range pruning during larger candidate atomset generation. The atoms that form the 9-atom substructure are mostly backbone atoms and β-carbon atoms in the helical region. Based on a comparison between the full structure and the algorithmically reformed helices, as shown in Figure 5, we conclude that this substructure is one of the basic structural units of the α-helix.



(a) 9-atom helical substructure (b) First two α-helices (c) Reconstructed α-helices

Figure 5: α helix reconstruction in Hemoglobin

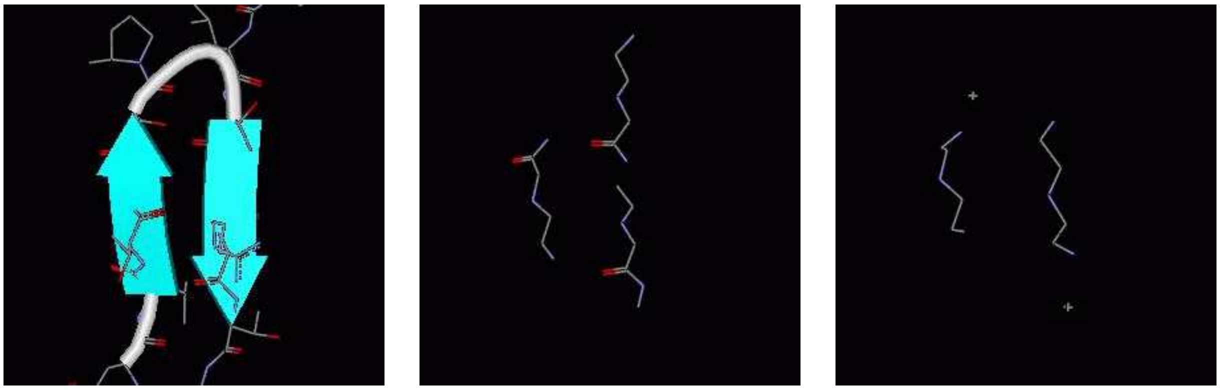


Figure 6: First β-Sheet (Left) and Reconstructed 8-Atom Strands (Center, Right) of Antibody 21D8

6.2.3 Discovery of β-Sheets

We complete our analysis of protein structure by discovering β-sheets. We examine the antibody Decarboxylase Catalytic Antibody 21D8-Hapten Complex (PDB ID: 1C5C), a two-chain β-sheet rich protein. We consider the first 73 residues of the first chain; this portion contains 3 sheets composed of 11 β-strands.

Since β-sheet structures are formed primarily from peptide-peptide interactions, we employ the backbone optimization for pre-processing the protein. This reduces the search space by a factor of 2. The algorithm is also seeded with parameters suitable for extended planar structures such as β-strands. A larger range of 6.5 $\mathring{A}$ is chosen to accommodate for the linear nature of these strands (compared to the tightly coiled nature of α-helices). We set the minimum support to 10 in an attempt to capture all 11 strands of interest. Lastly, we use a higher resolution, 0.1 $\mathring{A}$, since the coordinate resolution for this particular protein is worse than for previously mentioned proteins.

The algorithm locates five classes of frequent 8-atomsets; each of these classes describe different portions of the same substructure, which consists of three linear, planar, connected peptides. In other words, given the same 12-atom linear peptide sequence, the five classes describe different sets of 8 of these 12 atoms. These results are validated by the fact that β-strands have the same conserved structure as described above and consist of at least 3 amino acids.

The results when compared against the original antibody are impressive; every atomset found defines at least two resides in a β-strand, and 85% are fully in strands. Figure 6 shows the first two β-strands of Antibody 21D8 along with two different atomset representations of these β-strands.

6.2.4 Structural Similarity of Proteins

Last, we demonstrate how actual structural features, represented by substructure fingerprints, provide better insight into structure-function relationships than traditional sequence analysis. A substructure fingerprint for a particular protein is a vector representation of a set of interesting substructures; elements representing substructures contained in that protein are marked, either with a 1 if the simpler bit vector fingerprint is being constructed or with the number of substructures in the protein if a frequency vector fingerprint is being constructed.

To demonstrate the utility of substructure fingerprints, we consider the protein Ribonuclease A from two disparate species: bovine (PDB ID: 1JVT) and rat (PDB ID: 1RRA), as well as a similar protein from a related protein kinase class (PDB ID: 1BDY) and a significantly different protein, the ρ transcription terminator (PDB ID: 1A8V). The coordinate sets all have comparable resolution (between 2.0 and 2.5 $\mathring{A}$) and chain lengths (between 120 and 125 residues). Where multiple chains are present, we consider the first chain only.

We ran our algorithm on the above proteins, using the following parameters: 0.12 $\mathring{A}$ resolution, 6.5 $\mathring{A}$ range, minimum support of 20, and the backbone optimization. From the run, we collected all substructure motifs with five or more atoms and generated bit vector substructure fingerprints for the four proteins. We then analyzed fingerprints and sequences using the common Tanimoto similarity coefficient, defined as

$$Tanimoto(A,B) = \frac{\mid A \cap B \mid}{\mid A \mid + \mid B \mid - \mid A \cap B \mid}$$

The traditional sequence-based approach using the primary amino acid sequence gave only moderate similarity between the two Ribonuclease A proteins (Tanimoto coefficient of 0.50); this is a sign of evolutionary divergence and demonstrates the main limitation of sequence analysis. However, when comparing the Ribonuclease A substructure-based bit vector fingerprints, we obtain a Tanimoto similarity coefficient of *0.75*, which is a significant improvement over the sequence-based approach. This is to be expected, since the functional (and thus structural) characteristics of Ribonuclease A are evolutionarily conserved despite frequent sequence mutations.

Furthermore, sequence analysis of the two Ribonuclease A proteins with the kinase (functionally related) and transcription factor (functionally unrelated) proteins give Tanimoto coefficients of only 0.03, suggesting no evolutionary relationship. Yet our approach can detect substantial structural similarity between the Ribonuclease A and related kinase proteins (coefficient ranging from 0.35 to 0.45), which is consistent with the similar function of the proteins' classes. Furthermore, our approach detects little structural similarity (coefficient of 0.17) between the Ribonuclease A and transcription factor proteins, a result consistent with the disparity in function.

While preliminary, these results support the commonly held belief that a scalable structure-based technique such as ours is capable of more robust and efficient protein classification than traditional sequence-based approaches. We are actively working on development of more robust frequency-vector fingerprint techniques to further improve similarity metrics. Furthermore, we are researching the use of these similarity metrics to provide a more accurate classification system for proteins based on their three-dimensional characteristics.

7 Conclusion

This paper presents a novel approach and several key optimizations for mining frequent substructures in complex, noisy spatial data such as macromolecules. The approach differs from previous substructure mining techniques in that it locates frequent substructures *within a single large molecule* and is designed specifically to address scalability and noise issues chronic to the macromolecule domain. Furthermore, it differs from previous macromolecule data mining techniques in that it operates on exact structures instead of sequences or meta-structural information. Through a series of experiments, the algorithm is validated both for good performance when compared to standard techniques and for good frequent substructure identification as evidenced by its ability to detect peptide backbones, α-helices, and β-beta sheets in proteins as well as common structural features between similar proteins from different species.

The use of range pruning to reduce the chemical search space is a key optimization and is critical to the good scalability of our approach. In addition to restricting the search space, it also permits us to store less information per atomset, improving both memory efficiency and performance (since the number of mining bonds is fewer). We provide further reduction to the macromolecule search space by analyzing only the polymer backbone. While this approach does not capture side chain interactions, it does allow quick definition of the overall macromolecule structure. This definition can then be used for initial classification of proteins based on their general shape.

Recursive fuzzy hashing and distance binning minimize the effects of coordinate noise characteristic of crystallographic data, permitting detection of larger substructures where traditional techniques fail. Larger, more complex structural patterns are known to exist in proteins; these include particular meaningful combinations of α-helices and β-sheets as well as new structures, including binding sites such as the heme iron-binding structure of hemoglobin. With an efficient and scalable algorithm, discovery of novel super-structures like these can be automated; new super-structures would further scientists' understanding of the structure-function relationship.

Performance is always an issue when dealing with such large molecules, and as such research is in progress for further optimizing analysis based on both domain knowledge and computer science principles. One domain-centered approach for proteins is consideration of peptide ϕ and ψ angles in reduction of search space. More efficient approximate pruning, such as use of a three-dimensional sliding box, are under consideration, and the algorithm will soon be parallelized to further improve performance. Also, graph compression techniques using substructures [6] could be used to further optimize protein analysis.

With a framework in place for efficient analysis of substructures in macromolecules, we are now able to conduct further research in this area in a timely manner. Future work includes the analysis of higher-order substructures in an attempt to discover novel secondary structures and the development of classification models for proteins based on discovered substructures. In addition, we hope to extend this approach to other types of biologically significant macromolecules such as DNA and the various forms of RNA. Eventually, this algorithm will be combined with other data mining techniques to provide a robust structural analysis framework for spatial datasets.

References

[1] R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In *VLDB*, 1994.

[2] R. Agrawal and R. Srikant. Mining sequential patterns. In *ICDE*, 1995.

[3] R. Agrawal and R. Srikant. Mining sequential patterns: Generalizations and performance improvements. In *Proceedings of the Fifth International Conference on Extending Database Technology*, Mar 1996.

[4] E. G. Coffman and J. Eve. File structures using hashing functions. *Comm. Assoc. Comp. Mach.*, 1970.

[5] L. Dehaspe, H. Toivonen, and R. King. Finding frequent substructures in chemical compounds. In *The Fourth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1998.

[6] S. Djoko, D. Cook, and L. Holder. Analyzing the benefits of domain knowledge in substructure discovery. In *The First ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1995.

[7] J. Dougherty, R. Kohavi, and M. Sahami. Supervised and unsupervised discretization of continuous features. In *Machine Learning: Proceedings of the Twelfth International Conference*, San Francisco, 1995.

[8] X. Wang et. al. Automated discovery of active motifs in three dimensional molecules. In *The Third ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1997.

[9] A.K. Jain and R.C. Dubes. *Algorithms for Clustering Data*. Prentice Hall, 1988.

[10] I. Jonassen, I. Eidhammer, D. Conklin, and W. Taylor. Structure motif discovery and mining the pdb. In *German Conference on Bioinformatics*, 2000.

[11] J. Kim, E. Moriyama, C. Warr, P. Clyne, and J. Carlson. Identification of novel multi-transmembrane proteins from genomic databases using quasi-periodic structural properties. *Bioinformatics*, 2002.

[12] R. King, A. Karwath, A. Clare, and L. Dehaspe. Genome scale prediction of protein functional class from sequence using data mining. In *The Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2000.

[13] K. Koperski, J. Han, and N. Stefanovic. An efficient two-step method for classification of spatial data. In *Proceedings of the International Symposium on Spatial Data Handling*, Jul 1998.

[14] H. Li and S. Parthasarathy. Automatically deriving multi-level protein structures through data mining. In *HiPC Conference Workshop on Bioinformatics and Computational Biology*, Hyderabad, India, 2001.

[15] H. Mannila and H. Toivonen. Discovering generalized episodes using minimal occurrences. In *Second International Conference on Knowledge Discovery and Data Mining*, pages 146–151, Portland, Oregon, Aug 1996.

[16] W. Pan, J. Lin, and C. Le. Model-based cluster analysis of microarray gene-expression data. *Genome Biology*, 2002.

[17] S. Parthasarathy, M. Zaki, M. Ogihara, and S. Dwarkadas. Incremental and interactive sequence mining. In *CIKM*, 1999.

[18] J.R. Quinlan. Induction of decision trees. *Machine Learning*, 5(1):71–100, 1996.

[19] L. De Raedt and S. Kramer. The level-wise version space algorithm and its application to molecular fragment finding. In *Seventeenth International Joint Conference on Artificial Intelligence*, 2001.

[20] X. Zheng and T. Chan. Chemical genomics: A systematic approach in biological research and drug discovery. *Current Issues in Molecular Biology*, 2002.